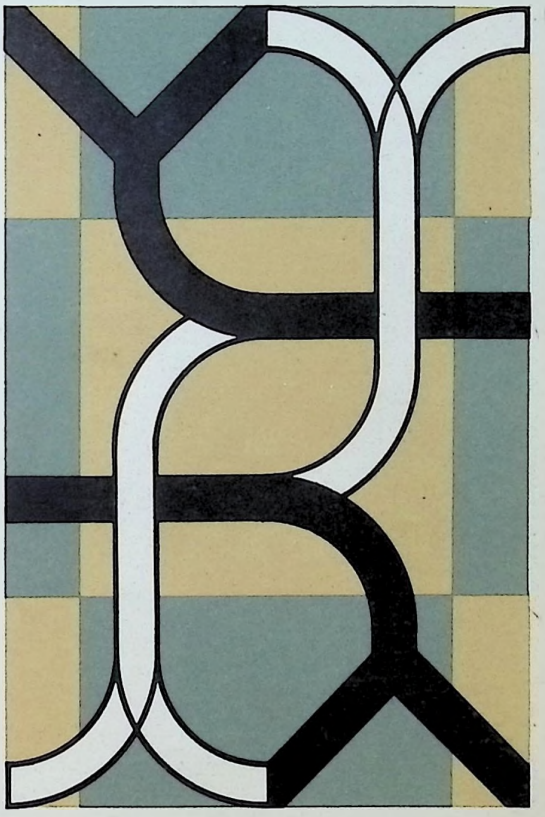
OLIVETTI

GENERAL ELECTRIC

Servizio Tecnico



CORSI DI ISTRUZIONE PROGRAMMATA

SISTEMA GE-115

1

INTRODUZIONE

4.615.0.500.0

CORSI DI ISTRUZIONE PROGRAMMATA

SISTEMA GE – 115

INTRODUZIONE

1

OLIVETTI
GENERAL ELECTRIC
Servizio Tecnico

Questo volume é stato realizzato dal
GRUPPO ISTRUZIONE PROGRAMMATA
della Scuola del Servizio Tecnico

PREFAZIONE

La presente pubblicazione é rivolta a coloro che si accingono allo studio dell'elaboratore GE - 115. Essa fa parte di una serie di volumi preparati con il Metodo della Istruzione Programmata.

Lo scopo di questo metodo é di fornire a chi studia gli elementi essenziali per l'autoapprendimento della materia esposta.
I volumi sono organizzati in lezioni sufficientemente brevi in relazione agli argomenti trattati. Ciò consente di fare acquisire pochi concetti elementari alla volta, ma tuttavia concatenati fra loro.

In ogni lezione, una o più volte, si cerca di focalizzare l'argomento o gli argomenti di maggiore interesse, proponendo all'interessato delle domande le cui risposte, a titolo di verifica, possono essere confrontate con quelle riportate nelle pagine seguenti.

Talvolta, se la complessità della materia lo richiede, l'interessato deve eseguire, o completare, degli esercizi in modo da poter integrare, con gli stessi, le nozioni acquisite.

Per evitare la possibilità di un apprendimento frammentario, che porterebbe alla perdita della visione di insieme, sono stati predisposti dei riepiloghi di carattere generale. Ciascun riepilogo riguarda la materia esposta nelle lezioni ad esso precedenti.
Tali riepiloghi sono stati preparati in modo da consentire all'interessato di verificare il proprio grado di preparazione.

Milano, giugno 1967

METODO DI STUDIO

Il metodo che si consiglia di seguire durante lo studio si articola in più punti.

E' necessario che l'interessato non prosegua ininterrottamente lo studio per tutto il tempo che ha a disposizione, anzi é utile che stabilisca alcuni intervalli; atti a ponderare quanto studiato, con una frequenza adeguata alle proprie capacità di assimilazione.

Ogni singola lezione va studiata integralmente ed in maniera non superficiale anche se qualche volta può sembrare di facile comprensione.

L'interessato deve cercare di dare sempre le risposte alle domande che gli sono proposte, prima di controllare le stesse con le risposte riportate.

Gli esercizi vanno svolti integralmente e possibilmente discussi con eventuali colleghi di studio.

Nel "Piano di studio", e all'inizio di ogni lezione é indicato il tempo necessario a svolgerla. Detto tempo ha solo uno scopo indicativo e generalmente é da ritenersi un tempo massimo.

Onde evitare la possibilità di un apprendimento frammentario, che porterebbe alla perdita della visione di insieme, sono stati predisposti dei riepiloghi di carattere generale. Essi consistono in una serie di domande alle quali é necessario rispondere scegliendo una delle risposte riportate sotto ciascuna domanda. Ogni risposta indica il numero della domanda successiva alla quale si deve rispondere e così via.

Se l'interessato risponde in maniera errata a qualche domanda viene inviato ad una successiva, tendente a chiarire il concetto non bene interpretato. Nel caso in cui, dopo uno o più tentativi, il lettore non riesca a trovare la via esatta viene rimandato a rivedere la lezione riguardante i concetti non bene appresi.

Durante lo svolgimento del riepilogo generale si consiglia di annotare il numero delle domande che vengono esaminate; ciò per facilitare il ritorno alla domanda dopo avere visto la lezione alla quale si era stati inviati.

Evidentemente, se durante lo svolgimento del riepilogo si risponde sempre in modo esatto a tutte le domande, al termine ne saranno state interessate solo un numero minimo. Tale numero minimo non é volutamente riportato nel testo.

Per poter abbandonare il questionario é necessario incontrare l'apposita indicazione.

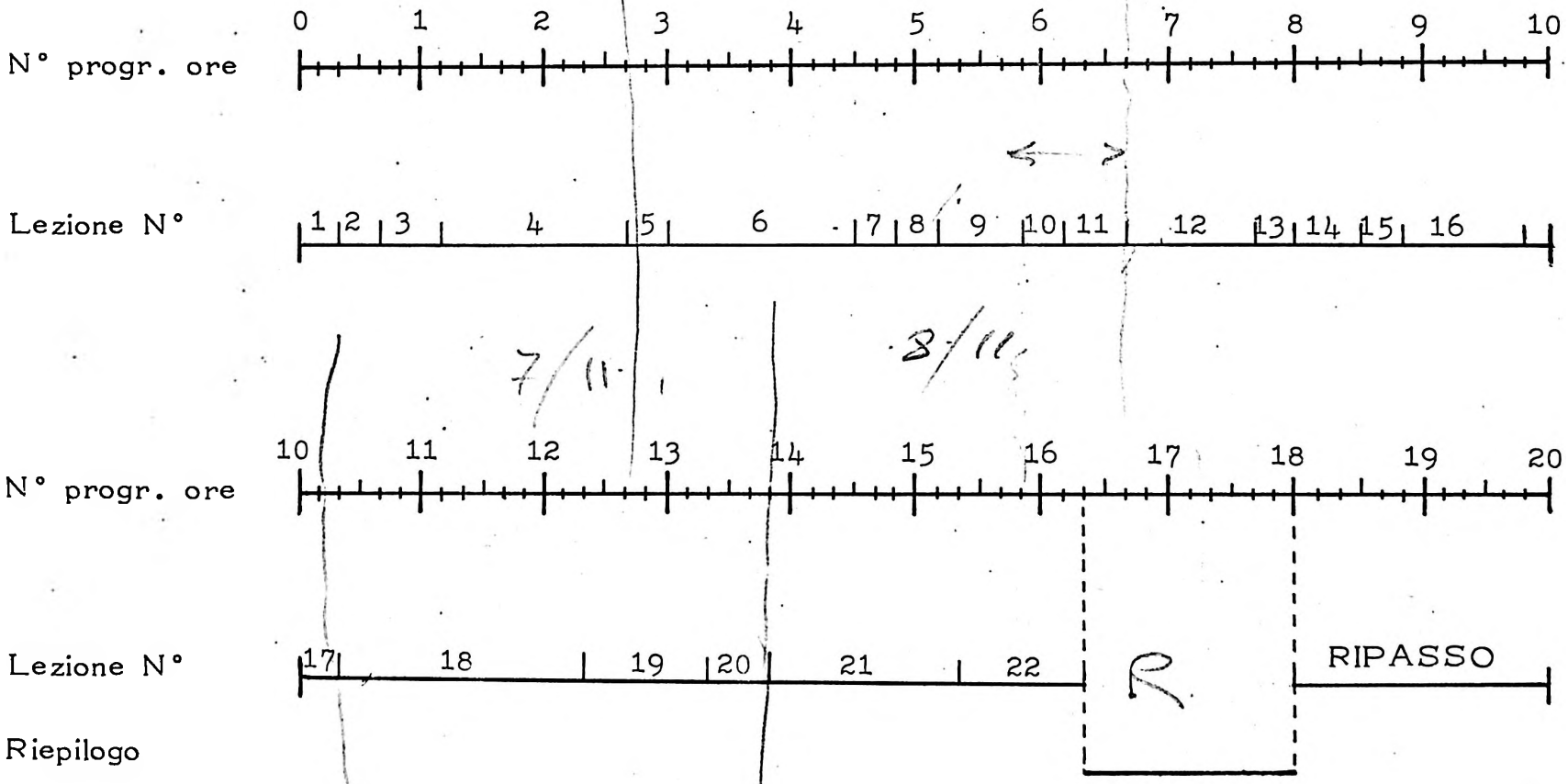
Al termine di ogni volume, il responsabile dello "Avanzamento Studio", proporrà all'interessato un questionario riguardante la materia trattata nel volume.

Al termine del corso teorico é previsto un corso pratico da effettuarsi alla Scuola Tecnici.

Per essere ammessi al corso pratico é necessario superare una prova teorica di carattere generale (tenuta alla Scuola) su tutto il programma svolto individualmente.

27-10 31-10 3/11

PIANO DI STUDIO



I N D I C E

	pagine
Prefazione	V
Metodo di studio	VII ÷ VIII
Piano di studio	IX
Indice	XI ÷ XII

Lezione 1 :	Generalità	1-3 ÷ 1-14
Lezione 2 :	Programma	2-3 ÷ 2-6
Lezione 3 :	Struttura di una istruzione	3-3 ÷ 3-12
Lezione 4 :	Codifica delle istruzioni	4-3 ÷ 4-24
Lezione 5 :	Istruzioni del GE-115	5-3 ÷ 5-12
Lezione 6 :	Istruzioni del gruppo A2	6-3 ÷ 6-19
Lezione 7 :	Caratteristiche comuni alle istruzioni del gruppo A1	7-3 ÷ 7-7
Lezione 8 :	Interpretazione di una istruzione del gruppo A1	8-3 ÷ 8-8

Lezione 9 :	Istruzioni del gruppo A1	9 - 3 ÷ 9 - 10
Lezione 10 :	Istruzioni del gruppo B3	10 - 3 ÷ 10 - 6
Lezione 11 :	Istruzioni del gruppo C4	11 - 3 ÷ 11 - 8
Lezione 12 :	Istruzioni di salto normale	12 - 3 ÷ 12 - 13
Lezione 13 :	Istruzioni di salto particolare	13 - 3 ÷ 13 - 6
Lezione 14 :	Istruzioni del gruppo D4	14 - 3 ÷ 14 - 12
Lezione 15 :	Istruzioni relative al pulsante STEP BY STEP	15 - 3 ÷ 15 - 6
Lezione 16 :	Analisi delle istruzioni	16 - 3 ÷ 16 - 22
Lezione 17 :	Fine fase alfa - Temporizzazione	17 - 3 ÷ 17 - 14
Lezione 18 :	Svolgimento di una istruzione	18 - 3 ÷ 18 - 23
Lezione 19 :	Studio di un programma	19 - 3 ÷ 19 - 33
Lezione 20 :	Interpretazione di programmi	20 - 3 ÷ 20 - 7
Lezione 21 :	Esame di un programma	21 - 3 ÷ 21 - 18
Lezione 22 :	Trascodifica	22 - 3 ÷ 22 - 23
Riepilogo		R - 3 ÷ R - 25

LEZIONE 1

Tempo di studio
20 minuti

1. GENERALITA'

1.1. DEFINIZIONE DI ELABORATORE

Lo scopo di questo volume é di introdurre nello studio del funzionamento e dell'utilizzazione dell'elaboratore GE - 115.

Per poter meglio comprendere ciò é opportuno analizzare prima alcuni concetti essenziali e alcune definizioni che ricorrono spesso nello studio di un elaboratore.

Si inizia con lo specificare che cosa si intende per elaboratore, quali sono le funzioni che deve svolgere, quali sono i compiti che deve eseguire.

Nel glossario edito dall" Executive office of the president bureau of the budget" del governo americano, l'elaboratore (computer in inglese) é definito nel seguente modo :

L'ELABORATORE E' UN APPARATO CAPACE DI ACCETTARE DELLE INFORMAZIONI, DI ESEGUIRE SULLE STESSE UN PREDETERMINATO PROCESSO E DI FORNIRE IN USCITA I RISULTATI DI QUESTO PROCESSO.

GENERALMENTE L'ELABORATORE E' DOTATO DI ORGANI DI INGRESSO, DI REGISTRAZIONE, DI USCITA DELLE INFORMAZIONI E DI ORGANI IN GRADO DI ESEGUIRE SIA OPERAZIONI LOGICHE CHE ARITMETICHE SULLE INFORMAZIONI STESSE.

A seconda del processo con il quale sono trattate le informazioni introdotte nell'elaboratore si possono classificare gli elaboratori in due diverse specie :

- elaboratori digitali
- elaboratori analogici

L'ELABORATORE DIGITALE E' UN ELABORATORE NEL QUALE LE INFORMAZIONI INTRODOTTE SONO RAPPRESENTATE SOLTANTO PER MEZZO DI DUE DIVERSI LIVELLI DI TENSIONE CHE PERMETTONO DI ESPRIMERE LE STESSE SOTTO FORMA DI BIT (BINARY DIGIT).

L'elaboratore analogico é un elaboratore nel quale le informazioni sono rappresentate per mezzo di analogie fisiche, ad esempio : temperatura, pressione, posizione angolare, valori di tensione, ecc. ogni diversa informazione può essere espressa con una diversa analogia fisica.

Gli elaboratori comunemente usati per la elaborazione dei dati sono, di solito, di tipo digitale.

Quelli analogici sono utilizzati nel campo della industria per la automazione dei processi produttivi; nel campo militare per risolvere complessi problemi di balistica e in campo scientifico nei casi in cui la complessità della materia lo richiede.

In questo volume, si esamineranno soltanto organi del GE - 115 che é un elaboratore digitale, per cui, ogni volta che si troverà il termine "elaboratore" si deve intendere elaboratore digitale.

1.2. SUPPORTI DELLE INFORMAZIONI

Nella definizione di elaboratore data nel precedente paragrafo, si é accennato ad organi di ingresso e ad organi di uscita.

In fig. 1.2.5 si é riportato uno schema di principio di una possibile configurazione dell'elaboratore GE - 115 nella quale é messo in evidenza il flusso delle informazioni, in ingresso e in uscita dalla unità centrale.

SI DEFINISCE UNITA' CENTRALE DI UN ELABORATORE L'INSIEME DEGLI ORGANI CHE ELABORANO LE INFORMAZIONI E CHE, FISICAMENTE, SONO RAGGRUPPATI IN UN UNICO COMPLESSO.

I supporti delle informazioni sono di varie specie :

- Scheda perforata, nella quale le informazioni sono espresse con perforazioni, secondo un certo codice.

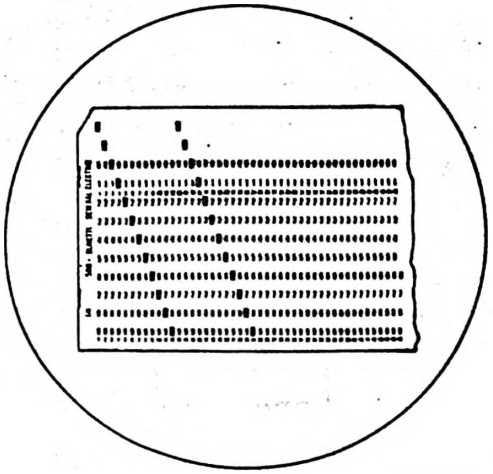
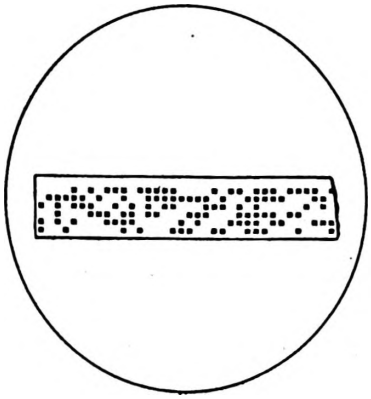


fig. 1.2.1

Questo supporto, insieme a quelli quì di seguito elencati, é utilizzato nel modo illustrato dallo schema di principio di fig. 1.2.6.

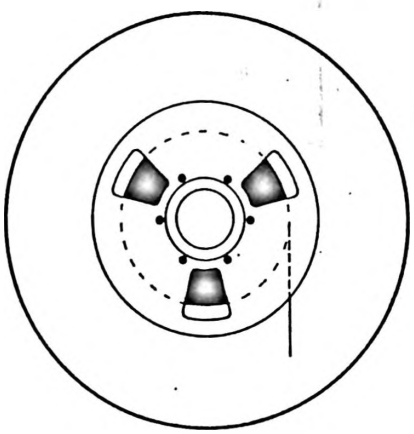
- Banda perforata, costituita da una striscia continua di carta lungo la quale sono perforate le informazioni secondo un certo codice

fig.1.2.2



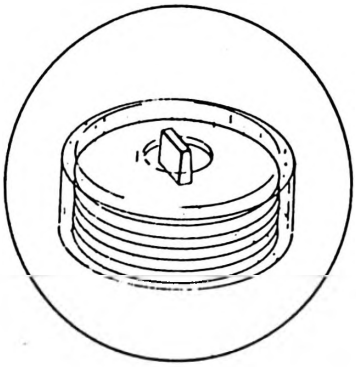
- Nastri magnetici, sui quali le informazioni sono registrate magnetizzando opportunamente la sostanza magnetica depositata su una faccia del nastro .
  Il nastro é raccolto in contenitori denominati :
  BOBINE :

fig. 1.2.3



- Dischi magnetici, sui quali le informazioni sono registrate magnetizzando opportunamente la sostanza magnetica depositata sulle facce di ogni disco :

fig. 1.2.4



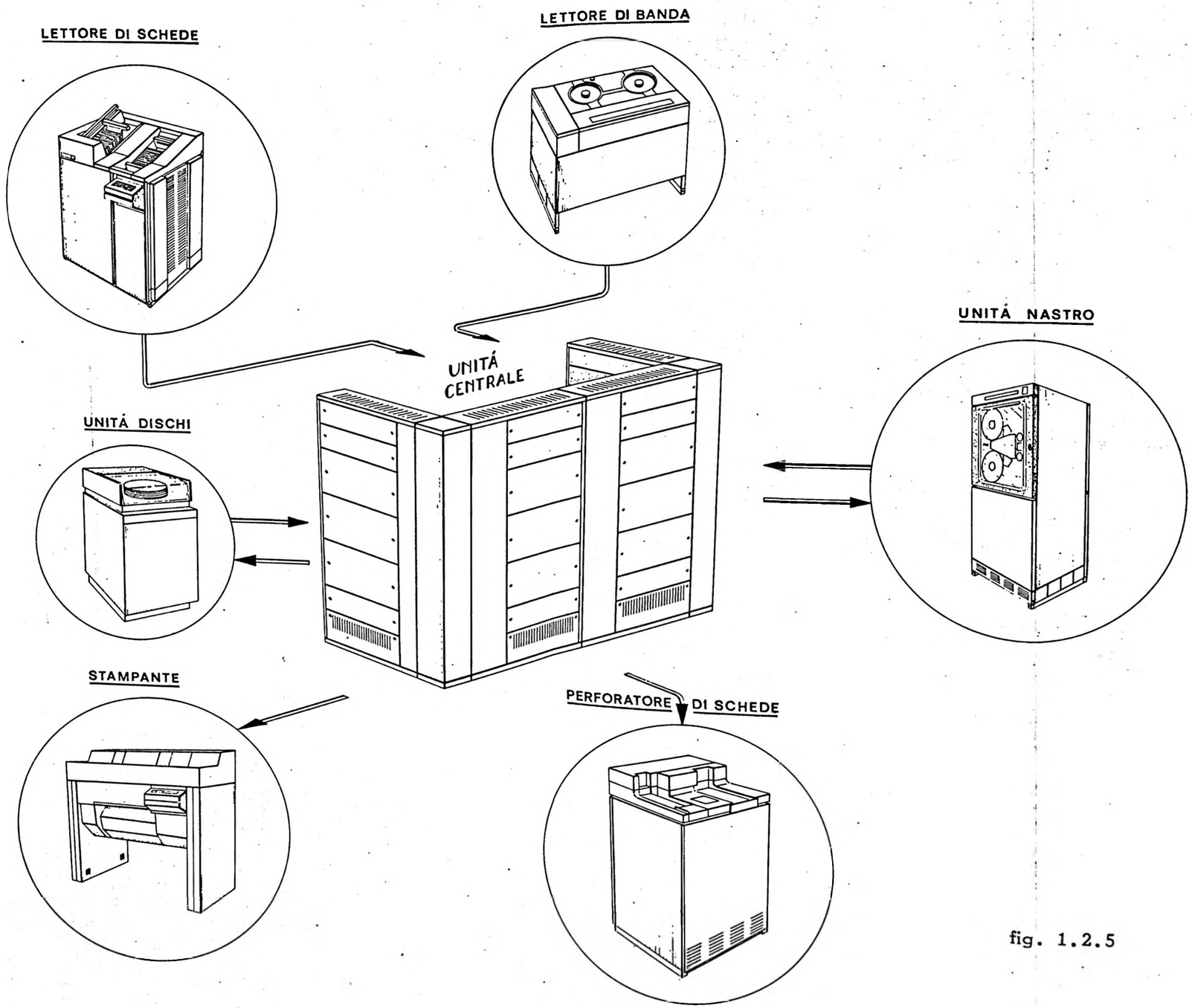

LETTORE DI SCHEDE
LETTORE DI BANDA
UNITÁ CENTRALE
UNITÁ DISCHI
UNITÁ NASTRO
STAMPANTE
PERFORATORE DI SCHEDE

fig. 1.2.5

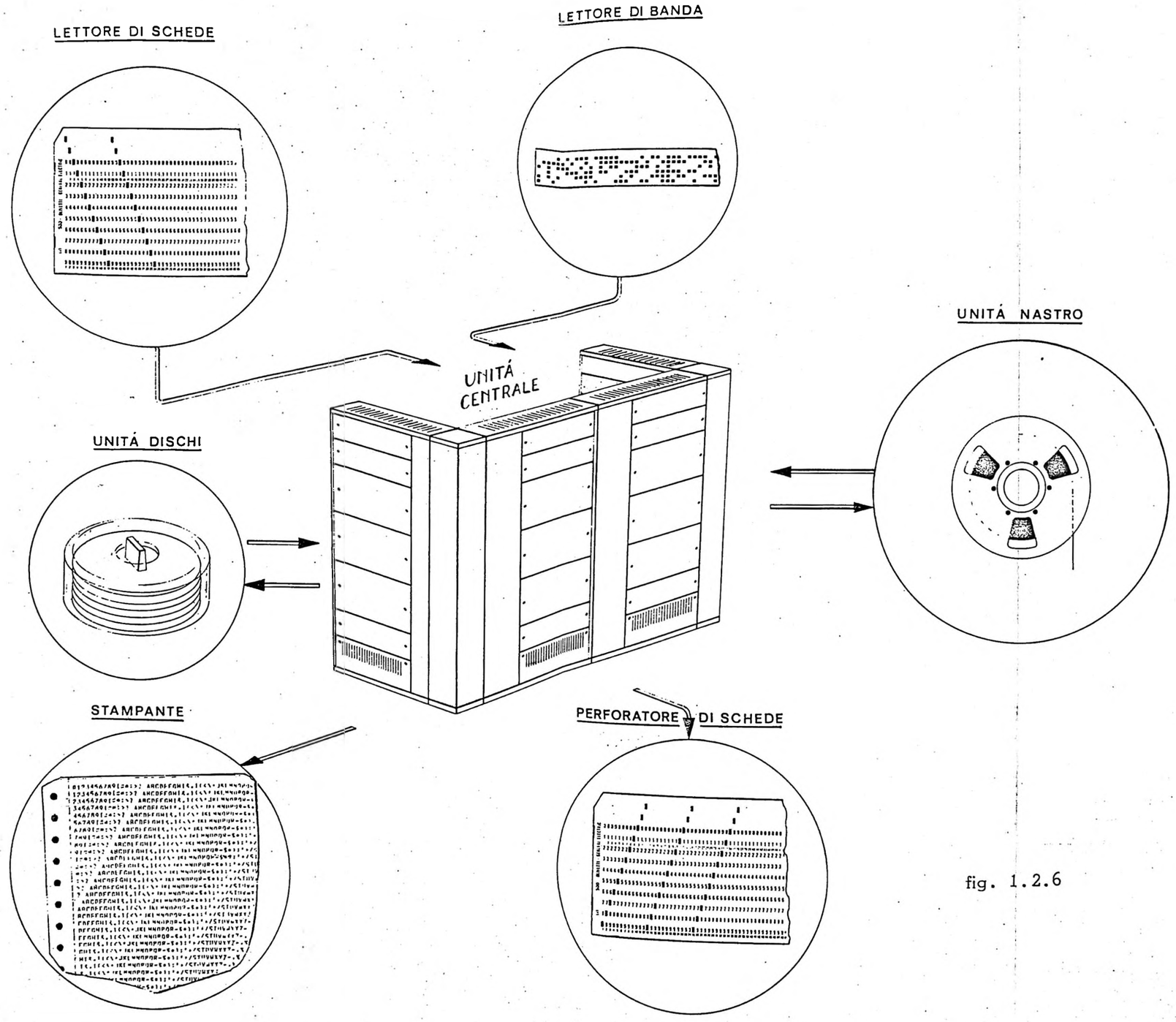

LETTORE DI SCHEDE
LETTORE DI BANDA
UNITÁ CENTRALE
UNITÁ NASTRO
UNITÁ DISCHI
STAMPANTE
PERFORATORE DI SCHEDE

fig. 1.2.6

1.3. UNITA' PERIFERICHE

Tutti i supporti menzionati al paragrafo 1.2. devono essere utilizzati mediante particolari apparecchiature.

Le informazioni da introdurre in un elaboratore possono essere lette da :

- LETTORE DI SCHEDE
- LETTORE DI BANDA
- UNITA' NASTRO MAGNETICO (lett. delle inform.)
- UNITA' DISCHI (lett.delle inform.)

Ognuna di queste apparecchiature, denominata genericamente UNITA' PERIFERICA, deve leggere informazioni registrate sui relativi supporti ed inviarle all'unità centrale.

Le unità periferiche sono dotate di un GOVERNO che coordina lo scambio di informazioni con l'unità centrale.

Le informazioni non sono inviate direttamente dall'unità periferica all'unità centrale bensì attraverso il GOVERNO.

SI DEFINISCE GOVERNO UN INSIEME DI ORGANI LOGICI, CHE PROVVEDONO ALLA COORDINAZIONE DELLO SCAMBIO DI DATI FRA UNITA' CENTRALE E UNA UNITA' PERIFERICA.

Le unità periferiche in grado di ricevere le informazioni in uscita della unità centrale sono :

- STAMPANTE
- PERFORATORE DI SCHEDE
- UNITA' NASTRO (registrazione delle informazioni)
- UNITA' DISCHI (registrazione delle informazioni)

Anche queste unità periferiche sono dotate di un governo che coordina lo scambio di informazioni.

Lo scopo di questo volume non é quello di studiare la costituzione e il funzionamento delle unità periferiche ed il relativo governo, (per le quali sono stati preparati appositi volumi), ma di studiare il principio di funzionamento della unità centrale.

1.4. REGISTRO

L'unità centrale riceve e fornisce informazioni, e pertanto deve essere dotata di organi in grado di contenere le stesse.

Con il termine generico di informazione sono stati sin quì indicati i caratteri espressi nel codice utilizzato dall'elaboratore, che costituiscono i dati da elaborare.

Il GE - 115 utilizza un codice a 8 bit le cui caratteristiche sono descritte in una lezione seguente.

L'unità centrale deve quindi avere degli organi in grado di contenere almeno 1 carattere, essi sono denominati REGISTRI.

IL REGISTRO E' UN ORGANO, REALIZZATO CON FLIP-FLOP, CAPACE DI CONTENERE UN CARATTERE QUALSIASI PER UN DETERMINATO TEMPO E DI RENDERLO DISPONIBILE PER ESSERE UTILIZZATO.

In fig. 1.4.1 é riportato un esempio di utilizzazione di un registro.

I caratteri sono letti dalla unità periferica interessata, sono introdotti, uno alla volta, nella unità centrale per essere staticizzati nel registro.

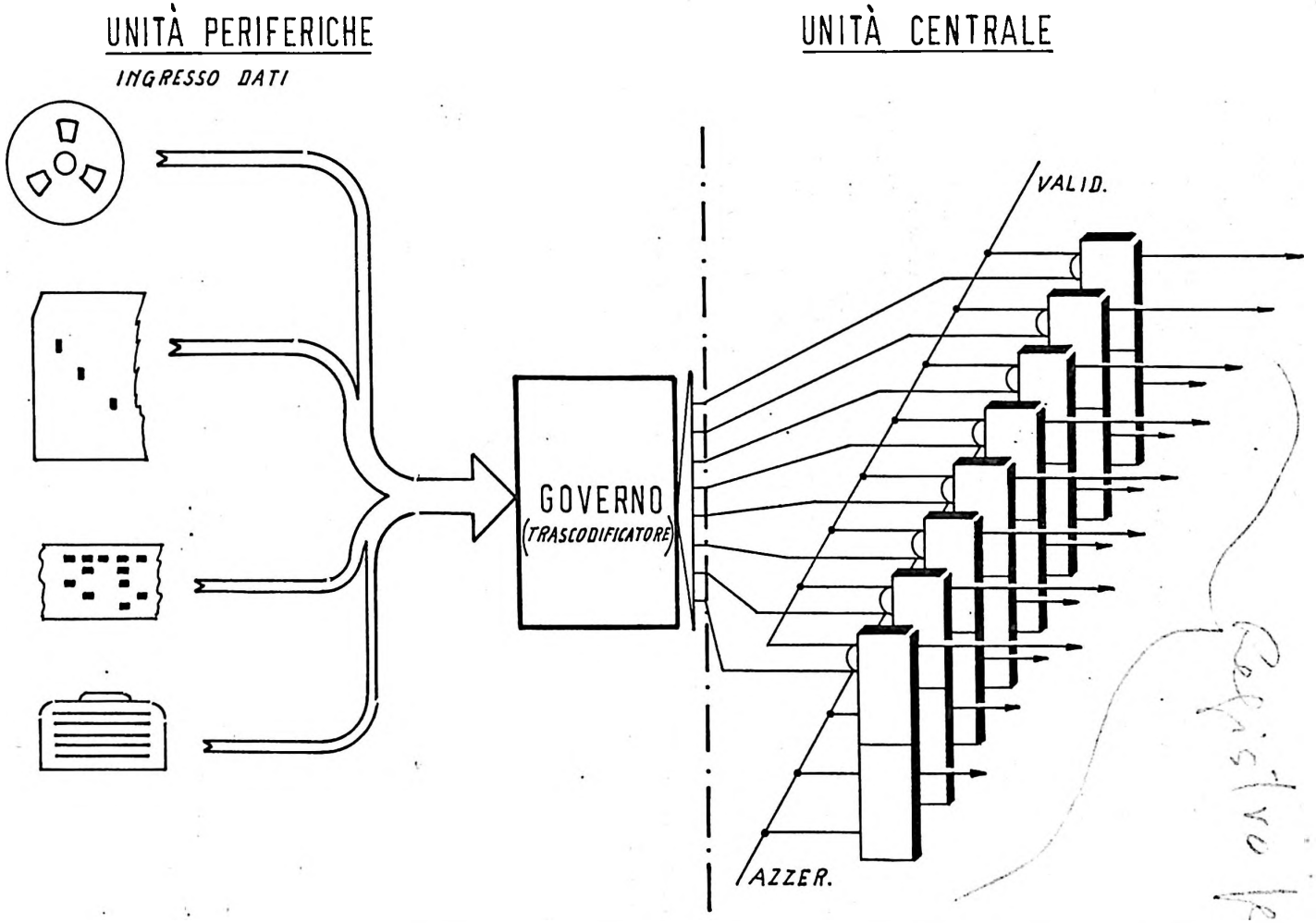

UNITÀ PERIFERICHE
INGRESSO DATI
UNITÀ CENTRALE
GOVERNO
(TRASCODIFICATORE)
VALID.
AZZER.

fig. 1.4.1

Normalmente l'introduzione di un carattere in unità centrale, avviene attraverso un organo denominato TRASCODIFICATORE, che ha lo scopo di "tradurre" i caratteri letti in un supporto, in un linguaggio comprensibile all'elaboratore, cioé nel codice utilizzato dallo stesso.

Nell'esempio di fig. 1.4.1 , il trascodificatore fornisce 8 segnali, corrispondenti agli 8 bit del caratttere trascodificato, che si presentano agli ingressi in maschera dei relativi flip-flop.

Quando gli ordini logici dell'elaboratore forniscono la validazione, (segnale indicato con la abbreviazione VALID), il carattere viene introdotto nel registro. Da questo istante le uscite del registro, forniscono il carattere ad altri organi dell'unità centrale, sino a quando il segnale AZZER non resetterà il registro stesso.

Ad esempio, il carattere staticizzato nel registro, può essere trasferito in una memoria a nuclei che ha la capacità di contenere contemporaneamente un elevato numero di caratteri.

LEZIONE 2

Tempo di studio
20 minuti

2. PROGRAMMA

2.1. L'unità centrale é composta da un insieme di organi. Nella precedente lezione si é visto l'esempio di un registro con i relativi segnali.

Ora é necessario conoscere come questi segnali sono generati, in quale modo gli stessi validano un organo anziché un altro : cioé é necessario conoscere tutte quelle operazioni elementari che l'unità centrale deve compiere per poter elaborare i dati in essa introdotti.

Per eseguire ognuna di queste operazioni l'elaboratore deve ricevere delle opportune indicazioni. Queste in dicazioni sono denominate ISTRUZIONI.

LE ISTRUZIONI SONO UN INSIEME ORDINATO DI CARATTERI CHE VENGONO INTRODOTTI NELL'ELABORATORE PER ESSERE OPPORTUNAMENTE INTERPRETATI DAGLI ORGANI LOGICI, DELLO STESSO.QUESTI ULTIMI SI COMPORTANO IN MODO DIVERSO A SECONDA DELLE CONFIGURAZIONI BINARIE ASSUNTE DAI CARATTERI CHE COMPONGONO LE ISTRUZIONI STESSE.

Le istruzioni che l'elaboratore può correttamente interpretare sono in numero fisso.

Il numero e i tipi di istruzione determinano la struttura logica dell'elaboratore.

L'istruzione permette di eseguire una operazione definita : essa consente lo svolgimento di una piccola parte dell'elaborazione, cioé del lavoro da compiere.

Per poter eseguire completamente un certo lavoro é necessario fornire all'elaboratore un insieme di istruzioni opportunamente ordinate.

Una successione ordinata di istruzioni, introdotta nella memoria dell'elaboratore, costituisce un programma.

IL PROGRAMMA E' UNA COMBINAZIONE DI ISTRUZIONI ELEMENTARI CHE L'ELABORATORE DEVE ESEGUIRE PER OTTENERE IL RISULTATO PRESTABILITO.

Il programma si definisce interno in quanto la sequenza delle istruzioni che lo determinano viene registrata prima di iniziare la elaborazione nello stesso organo (memoria) che immagazzina i dati all'interno del l'elaboratore stesso.

L'elaboratore GE - 115 é a programmazione interna automatica; cioé ogni singolo programma esegue, una o più volte automaticamente, una parte o l'intera sequenza delle istruzioni elementari, fino ad ottenere i risultati previsti.

DOMANDE

a) - Le istruzioni correttamente interpretate da un elaboratore possono essere di qualsiasi tipo ?

sono quelle in cui l'elaborazione è stata prefissata

b) - Cos'é l'istruzione ?

Un insieme ordinato di caratteri sulla quale la elaboratore riconosce

c) - Da cos'é costituita un'istruzione ?

da un insieme di caratteri opportunamente disposti

d) - Da cos'é definita la struttura logica dell'elaboratore ?

dal N° e dal tipo di istruzioni

e) - Cosa s'intende per programma ?

Un insieme ordinato di istruzioni la quale un elaboratore esegue

f) - Perché si definisce programma interno ?

perché le istruzioni sono registrate nell'interno dell'elaboratore

g) - Perché si definisce programma interno automatico ?

perché il programma interno legge ed interpreta automaticamente la sequenza di operazioni che costituiscono il programma.

RISPOSTE

a) - No. L'elaboratore può interpretare correttamente solo le istruzioni per le quali é stato progettato.

b) - L'istruzione é un insieme ordinato di caratteri che hanno lo scopo di far svolgere all'elaboratore l'operazione elementare per la quale é stata pro - gettata.

c) - Da un insieme di caratteri opportunamente predi sposti.

d) - La struttura logica dell'elaboratore é definita dal numero e dai tipi di istruzioni.

e) - Per programma si intende un insieme di istruzio ni organizzate in modo da far svolgere all'elabora tore un determinato lavoro.

f) - Si definisce programma interno poiché é immagaz zinato nell'interno dell'elaboratore.

g) - Perché la programmazione interna automatica con sente di far svolgere automaticamente la sequen- za delle operazioni che determinano il programma.

LEZIONE 3

Tempo di studio
30 minuti

3. STRUTTURA DI UNA ISTRUZIONE

3.1. IL CARATTERE DI FUNZIONE

L'elaboratore GE - 115 ha la possibilità, di eseguire determinate operazioni. Queste sono definite da "ordini successivi" che l'elaboratore riceve analizzando le istruzioni.

Poiché le istruzioni sono tutte diverse tra loro, esiste un carattere tra quelli che le compongono, che ha lo scopo di distinguerle.

IL CARATTERE CHE DISTINGUE UN'ISTRUZIONE DA TUTTE LE ALTRE E' DENOMINATO CARATTERE DI FUNZIONE DELL' ISTRUZIONE.

Quindi si hanno tanti caratteri di funzione quante sono le istruzioni che l'elaboratore può interpretare correttamente.

Ogni carattere di funzione, detto anche codice di funzione, é diverso dagli altri.

DOMANDE

a) - Quanti sono i codici di funzione ?
tutte quante le istruzioni

b) - Cos'é il codice di funzione ?
l'elemento distintivo delle funzioni

c) - Quanti codici di funzione ci sono uguali tra di loro?
nessuno

d) - Qual'é l'elemento distintivo delle istruzioni ?
il codice di funzione

RISPOSTE

a) - Quante sono le istruzioni.

b) - E' il carattere che distingue le istruzioni tra di loro.

c) - Nessuno, perché proprio per distinguere le varie istruzioni occorre che i rispettivi codici di funzione siano tutti diversi fra loro.

d) - Il codice di funzione.

3.2. CAMPI

Alcune istruzioni per le quali l'elaboratore é stato progettato possono interessare una o due zone di memoria.

L'elaboratore può, per esempio, fare un trasferimento di caratteri : ciò significa spostare gli stessi da una zona ad un'altra di memoria

LE ZONE DI MEMORIA INTERESSATE DA UN' ISTRUZIONE SI DEFINISCONO CAMPI.

Nell'istruzione stessa é indicato da quale posizione si inizia a prelevare i caratteri e in quale posizione si inizia a registrarli dopo l'elaborazione.

Tra i caratteri che definiscono l'istruzione alcuni determinano l'inizio dei campi interessati dalla stessa.

Nelle istruzioni si indicano sempre gli indirizzi iniziali dei campi.

NELL' ELABORATORE GE - 115 I CAMPI VENGONO CHIAMATI "CAMPO E" E "CAMPO A"

Il CAMPO E nel caso in cui la istruzione lo richieda é il campo che VIENE MODIFICATO mentre il CAMPO A RESTA SEMPRE INALTERATO.

A seconda del tipo di istruzione possono essere interessati uno, due o nessun campo.

DOMANDE

a) - Cosa s'intende per campo ?
.....

b) - Da che cosa sono determinati i campi ?
.....

c) - Quanti campi si stabiliscono nel GE - 115 ?
.....

d) - L'utilizzazione dei campi differisce a seconda del tipo di istruzione che si svolge ?
.....

RISPOSTE

a) - Per campo s'intende una zona di memoria.

b) - Da dei caratteri che compaiono nell'istruzione.

c) - Due : E ed A.

d) - Si. Possono essere utilizzati entrambi contemporaneamente, oppure uno solo oppure nessuno.

3.3. LUNGHEZZE

I caratteri da elaborare, letti da una unità periferica, entrano in una zona di memoria quando sono richiesti dall'istruzione corrispondente.

Il numero di caratteri introdotti può essere variabile e, gli stessi, possono essere elaborati tutti o solo una parte.

Nel paragrafo precedente si é visto che occorre esprimere nella istruzione l'indirizzo iniziale dei campi (o del campo) interessati dalla elaborazione.

Per definire completamente una istruzione non é sufficiente indicare il carattere di funzione e gli indirizzi iniziali dei campi, ma occorre anche specificare il numero dei caratteri da elaborare.

IL NUMERO DI CARATTERI CHE SI VUOLE ELABORARE DEFINISCE LA LUNGHEZZA DI UN CAMPO CHE NORMALMENTE E' IL CAMPO E.

Il numero che definisce la quantittà dei caratteri da elaborare costituisce un carattere dell'istruzione stessa.

DOMANDE

Si abbia un campo così composto :

X -5 -7 -B -3 -8 -4 -3 -9 -2 -7 -6

caratteri

e si voglia effettuare il trasferimento dei caratteri 8 - 4 - 3 - 9 in un altro campo.

a) - Quale valore assume il carattere che esprime la lunghezza del dato 8439?

3

b) - Cosa s'intende per lunghezza ? Il N° dei caratteri del dato che si vuole elaborare

c) - Con che cosa si stabilisce la lunghezza di una operazione ? In l'es. nel trasferimento di cui sopra con il carattere 3.

che deve comparire nell'istruzione -

RISPOSTE

a) - 3. Poiché si conta a partire da zero.

b) - Il numero dei caratteri del dato sui quali si vuole elaborare.

c) - Con un carattere che compare nell'istruzione.

3.4. CASI PARTICOLARI

I codici che abbiamo esaminato si riferiscono ad una istruzione tipo.

Talora necessita stabilire non solo la lunghezza di un campo, bensì la lunghezza dei due campi : E ed A. Si vedrà in seguito come l'elaboratore si comporta esaminando un'istruzione di questo tipo.

E' necessario porre quì in evidenza un altro particolare relativo alle istruzioni.

Nelle istruzioni di salto l'operazione é vincolata al verificarsi, o al non verificarsi, di una o più condizioni particolari di macchina.

Le condizioni di macchina sono numerose e differenti tra loro; é indispensabile, quindi che all'elaboratore sia indicato a quale delle tante possibili condizioni va riferita l'istruzione di salto in esame.

Si introduce, a questo scopo, un codice (al posto di un altro) che definisca per quale condizione il salto deve verificarsi.

DOMANDE

a) - Il codice di lunghezza esprime sempre la stessa cosa in tutte le istruzioni?

.....

.....

b) - In quali istruzioni si sostituisce un codice con un altro che indica una condizione da verificare ?

.....

.....

RISPOSTE

a) - No, in alcune istruzioni esprime la lunghezza riferita ad un solo campo, in alcune altre esprimono la lunghezza riferita ai due campi.

b) - Nelle istruzioni di salto.

3.5. INDIRIZZI

La memoria utilizzata nell'elaboratore GE - 115 ha una capacità variabile da 4096 a 16384 posizioni.

In ogni posizione si può immagazzinare un solo carattere numerico od alfabetico.

Nella memoria si introduce sia il programma che i dati.

IL PROGRAMMA OCCUPA UNA CERTA ZONA DI MEMORIA CHE NON DEVE COINCIDERE CON QUELLA NELLA QUALE SI INTRODUCONO I DATI.

Le posizioni di memoria sono indicate a partire dallo zero per numeri crescenti, in relazione alla capacità totale.

Una memoria di 4096 posizioni ha la prima posizione indicata con lo zero e l'ultima con 4095.

Nella memoria a 16384 posizioni, la prima posizione é identificata con lo zero e l'ultima 16383.

Ogni numero compreso tra 0 e 16383 identifica una posizione di memoria e viene denominato indirizzo.

Quindi per indirizzo si definisce una posizione di memoria. Ad esempio, l'indirizzo 2995 identifica la ~~2956~~ 2996ª posizione di memoria.

DOMANDE

a) - Cosa indica un indirizzo di memoria ?
una posizione di Memoria

b) - Quanti caratteri si possono immagazzinare in una posizione di memoria ?
1

c) - Che capacità può avere la memoria del GE - 115 ?
.....4096 o 16384 posizioni.....

d) - Qual'é la differenza esistente tra un indirizzo e la corrispondente posizione di memoria ?
.....
indirizzo + 1 = posiz Memoria interessata

(Così → 0 1 2 3 ecc
1° pos 2° pos 3° pos)

RISPOSTE

a) - L'indirizzo indica una posizione di memoria.

b) - Uno ed uno soltanto.

c) - Da 4096 a 16384.

d) - L'indirizzo esprime la posizione di memoria diminuita di uno. (L'indirizzo 3 2 esprime la trentatreesima posizione di memoria).

LEZIONE 4

Tempo si studio
ore 1 e 30 minuti

4. CODIFICA DELLE ISTRUZIONI

4.1. CODICI DI UNA ISTRUZIONE IDEALE

Un'istruzione é composta da un certo numero di caratteri particolari che devono indicare all'elaboratore tutti gli elementi necessari per eseguire correttamente le operazioni elementari che si vogliono ottenere dall'istruzione stessa.

I caratteri che definiscono una istruzione tipo sono denominati :

- codice di funzione
- lunghezza dell'operazione (N° di car. da elabor.)
- indirizzo iniziale dei campi.

Tutti gli altri caratteri che sono introdotti nella memoria dell'elaboratore sono denominati DATI o meglio :

SI DEFINISCE DATO QUEL CARATTERE, O QUEL GRUPPO DI CARATTERI, CHE SONO INTRODOTTI IN MEMORIA PER ESSERE ELABORATI.

Si voglia ora eseguire il trasferimento del dato 187465 che si trova in memoria a partire dall'indirizzo 2000 e lo si voglia trasferire a partire dall'indirizzo 4000.

Indirizzi	2000	2001	2002	2003	2004	2005
Dato	1	8	7	4	6	5

Il primo carattere dell'istruzione deve indicare che l'istruzione stessa é di trasferimento. Esso é denominato codice di funzione e nel caso in esame si suppon ga che sia costituito dal carattere T.

All'elaboratore deve essere poi indicato anche la lunghezza del dato sul quale si vuole operare. (Numero dei caratteri da elaborare meno 1).

Poiché tale dato é 187465, la sua lunghezza, é determinata dal totale delle posizioni occupate in memoria meno uno, cioe 6 - 1 = 5 (2005 - 2000 = 5).

Infine, all'elaboratore deve essere indicato da quale posizione iniziare a prelevare il dato (indirizzo 2000) e in quale posizione iniziare a metterlo. (indirizzo 4000).

Da quanto espósto la nostra istruzione tipo, posta a partire dall'indirizzo 0000 di memoria, si presenta nel seguente modo :

0000	0001	0002	0003	0004	0005	0006	0007	0008	0009
T	5	2	0	0	0	4	0	0	0
F	L	Campo di Prelev.				Campo di Trasferim.			

Si può notare che nel caso considerato i caratte - ri che formano l'istruzione sono 10.

Esattamente :

1 (F) per esprimere il codice di funzione (T)

1 (L) per esprimere la lunghezza del dato (5)

4 per esprimere l'indirizzo iniziale del campo di prelevamento (2000)

4 per esprimere l'indirizzo iniziale del campo di trasferimento (4000).

DOMANDE

a) - Quanti caratteri sono necessari per definire una istruzione espressa in caratteri decimali ?

Max - f + C + minimo (4 bit | x 2)

4 of 12 Max

b) - Quante posizioni di memoria occupano i caratteri di detta istruzione ?

10 in l'esempio fatto

RISPOSTE

a) - Dieci caratteri.

b) - Dieci posizioni di memoria.

4.2. ANALISI DELL'ISTRUZIONE

Si supponga di avere ancora in memoria a partire dalla posizione 0000 l'istruzione precedentemente vista.

	1	2	3	4	5	6	7	8	9	10
Indir. mem.	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009
Car. istruzione	T	5	2	0	0	0	4	0	0	0

L'ELABORATORE INIZIA L'ANALISI DELL'ISTRUZIONE SEMPRE DALLA POSIZIONE DI MEMORIA DI INDIRIZZO MINORE E PROCEDE POI PER INDIRIZZI CRESCENTI.

Ciò significa che, nell'esempio fatto, legge prima il carattere T, poi il 5 e poi tutte gli altri caratteri fino alla fine.

Si vedrà in seguito come ha termine l'esame della istruzione.

Sia noto che quando é stato letto l'ultimo carattere, l'elaboratore ha terminato la fase di interpretazione dell'istruzione e dovrà iniziare quella di esecuzione.

DOMANDE

a) - Da quale indirizzo l'elaboratore inizia ad esaminare l'istruzione?
..... dall'ind. 0000 (il più basso) .....

b) - Come procede poi l'analisi ? In ordine crescente di indirizzo
.....

c) - Terminata la fase di interpretazione della istruzione quale altra fase ha inizio ?
..... la fase di esecuzione .....

RISPOSTE

a) - Dall'indirizzo minore.

b) - Per indirizzi crescenti.

c) - La fase di esecuzione.

4.3. POSIZIONI DI MEMORIA OCCUPATE DALL'ISTRUZIONE

L'istruzione in esame occupa 10 posizioni di memoria.

Se si considera che ogni istruzione occupa tante posizioni quanti sono i caratteri che la costituiscono, risulta evidente che tanto maggiore é il numero di tali caratteri per istruzione maggiori sono le posizioni occupate da un programma in memoria e minore é il numero delle posizioni che si lasciano libere per i dati.

PER DIMINUIRE IL NUMERO DELLE POSIZIONI OCCUPATE DA UN PROGRAMMA, E' INDISPENSABILE DIMINUIRE IL NUMERO DI CARATTERI CHE COMPONGONO UN'ISTRUZIONE.

Per meglio evidenziare la necessità di usare pochi caratteri per formare una istruzione é stato utilizzato un esempio di una istruzione solamente teorica.

In realtà le istruzioni non vengono espresse nella maniera esposta bensì si adottano accorgimenti particolari che portano alla riduzione del numero di caratteri che le compongono, nel modo descritto nei paragrafi seguenti.

DOMANDE

a) - Da che cosa dipende il numero delle posizioni di memoria occupate dal programma ?

dal N. di caratteri che compongono le istruzioni

b) - Da che cosa dipende il numero delle posizioni di memoria disponibili per i dati ?

dalla grandezza di posizioni di memoria lasciate libere dalle istruzioni o dal programma

RISPOSTE

a) - Dal numero delle istruzioni che costituiscono il programma e dal numero dei caratteri che compongono ogni singola istruzione.

b) - Dalle posizioni di memoria che rimangono libere dopo la introduzione del programma.

4.4. CODICE DI MACCHINA

Prima di analizzare la struttura delle istruzioni del GE - 115, é necessario esaminare il tipo di codice utilizzato.

IL CODICE USATO NEL GE - 115, DENOMINATO CODICE GECD, E' UN CODICE A 8 BIT. ESSO E' COSI FORMATO :

				h	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
				g	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
				f	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
				e	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	d	c	b	a																
	0	0	0	0					0						Λ	+				
1	0	0	0	1					1	A					J	/				
2	0	0	1	0					2	B					K	S				
3	0	0	1	1					3	C					L	T				
4	0	1	0	0					4	D					M	U				
5	0	1	0	1					5	E					N	V				
6	0	1	1	0					6	F					O	W				
7	0	1	1	1					7	G					P	X				
8	1	0	0	0					8	H					Q	Y				
9	1	0	0	1					9	I					R	Z				
10	1	0	1	0					[	&					-	\				
11	1	0	1	1					#	.					$	,				
12	1	1	0	0					_	]					*	%				
13	1	1	0	1					:	(					)	=				
14	1	1	1	0					>	<					;	"				
15	1	1	1	1					?	\					'	!				

Le zone circoscritte con linea grossa sono denominate SET GRAFICO o SOTTOINSIEME GRAFICO.

IL SET GRAFICO IDENTIFICA LE COMBINAZIONI BINARIE DEL CODICE CHE CORRISPONDONO AI CARATTERI STAMPABILI.

Tutte le altre combinazioni sono utilizzate per scopi diversi.

Come in seguito sarà illustrato, alcune di queste combinazioni sono utilizzate per formare i caratteri che compongono le istruzioni (carattere di funzione, ecc).

4.5. METODO ESADECIMALE

Nel paragrafo precedente é stato presentato il codice a 8 bit GECD.

La lettura e la scrittura di una combinazione binaria a 8 bit non é molto agevole per cui si é ricorsi a degli artifici per esprimere in modo più semplice un carattere.

I bit del codice sono stati suddivisi in due gruppi:

- un gruppo che comprende i quattro bit più significativi (e,f,g,h)
- un gruppo che comprende i quattro bit meno significativi (a,b,c,d).

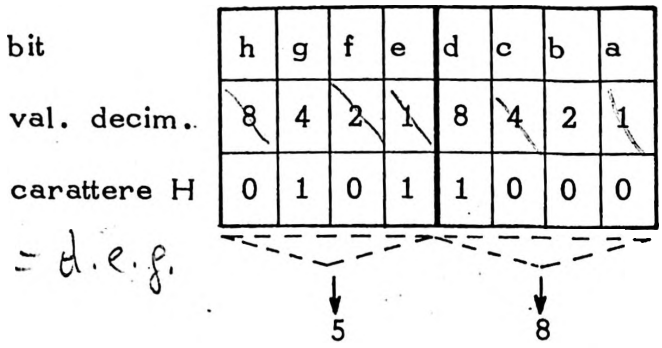
Ad ognuno di questi gruppi si assegna convenzionalmente il valore decimale 15.

bit	h	g	f	e	
valore decimale	8	4	2	1	= 15
bit	d	c	b	a	
valore decimale	8	4	2	1	= 15

In questo modo un carattere del codice GECD, ad esempio il carattere H, é scritto con i valori deci

mali, dei due gruppi di bit, separati da una barra :

5/8



Un altro esempio: il carattere V = 11/5.

Il metodo ora descritto é denominato METODO ESADECIMALE.

Per i caratteri che hanno i due gruppi di bit con valore decimale superiore a nove, cioé da 10 a 15 compreso, si indica il valore del gruppo con una lettera anziché con due cifre.

Si attribuisce la lettera

A	al	valore	decimale	10
B	"	"	"	11
C	"	"	"	12
D	"	"	"	13
E	"	"	"	14
F	"	"	"	15

In questo modo il carattere V che più sopra era stato scritto 11/5, viene ora scritto B/5.

Questo metodo di rappresentazione esadecimale dei caratteri, con lettere per i valori 10 ÷ 15, é denominato ESADECIMALE CODIFICATO.

DOMANDE

a) - Quale configurazione binaria ha il carattere "!" del codice GECD ?

h	g	f	e	d	c	b	a
1	0	1	1	1	1	1	1

b) - Come si scrive il carattere "!" con il metodo esadecimale ed esadecimale codificato ?

11 / 15 — B-F

.....

.....

c) - Che cosa é il set grafico ?

la zona... caratteri

.....

Simboli... la stampa ec

RISPOSTE

a) -

h	g	f	e	d	c	b	a
1	0	1	1	1	1	1	1

b) - 11/15 ; B/F

c) - Il set grafico é l'insieme delle combinazioni binarie del codice utilizzato dall'elaboratore corrispondente all'insieme dei caratteri stampabili.

4.6. INDIRIZZI DI MEMORIA ESPRESSI IN UNA ISTRUZIONE

Nella prima istruzione analizzata nei paragrafi precedenti gli indirizzi d'inizio campo sono stati espressi direttamente in valore decimale.

Si é tuttavia constatato che le posizioni di memoria occupate dall'istruzione considerata sono numerose e che possono essere ridotte adottando opportuni accorgimenti.

L'accorgimento impiegato nel GE - 115 é quello di esprimere il valore degli indirizzi non in decimale, ma in valore binario.

Si supponga di avere una memoria di 256 posizioni : in questo caso, con un codice a 8 bit ogni singola posizione può essere identificata da una configurazione delle 256 possibili ottenute dalla combinazione dei bit (2^8).

Per esempio l'indirizzo 0000 di memoria sarà espresso :

bit	07	06	05	04	03	02	01	00
valore binario	0	0	0	0	0	0	0	0
esadec.	0				0			

L'indirizzo 215 é espresso con i caratteri D/7 :

bit	07	06	05	04	03	02	01	00
valore binario	1 (128)	1 (64)	0	1 (16)	0	1 (4)	1 (2)	1 (1)
esadec.	D				7			

e l'indirizzo 255 con F/F :

bit	07	06	05	04	03	02	01	00
valore binario	1_{128}	1_{64}	1_{32}	1_{16}	1_{8}	1_{4}	1_{2}	1_{1}
esadec.	F				F			

255

Nell'ipotesi fatta ci si riferiva ad una memoria di 256 posizioni in realtà una memoria di questa capacità é insufficiente in quanto troppo ridotta.

Il GE-115 ha una memoria di 4096 posizioni corrispondente al binario 2^{12}.

Per ottenere i valori corrispondenti alle configurazioni di 2^{12} sono quindi indispensabili 12 bit cioé 4 in più di quelli precedentemente considerati.

I bit supplementari vengono chiamati 08-09-10-11 e sono aggiunti ai precedenti otto.

Ciò dà la possibilità di ottenere 4096 configurazioni in quanto :

bit	11	10	09	08	07	06	05	04	03	02	01	00
valore decim.	2048	1024	512	256	128	64	32	16	8	4	2	1

Essendo però prevista per il GE - 115, anche una capacità di memoria di 16384 posizioni é necessario una aggiunta di bit che possa creare ulteriori configurazioni.

Poiché $16384 = 2^{14}$ si dovranno avere 14 bit per esprimere uno qualsiasi dei possibili indirizzi, e cioé :

bit	13	12	11	10	09	08	07	06	05	04	03	02	01	00
valore decim.	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

8 bit = 256 ps

DOMANDE

a) - Possono i soli bit del codice macchina che sono 8 dare 1725 configurazioni ?

NO

b) - Qual'é la capacità minima di memoria ?

4096 = 12 bit

c) - Quanti bit occorrono per poter interessare tutte le 4096 posizioni di memoria ?

12

d) - Quanti bit occorrono per poter interessare tutte le posizioni di una memoria di capacità massima ?

16 bit

RISPOSTE

a) - No, poiché il codice può dare solo 256 configurazioni diverse.

b) - 4096 posizioni.

c) - 12, in quanto $2^{12} = 4096$

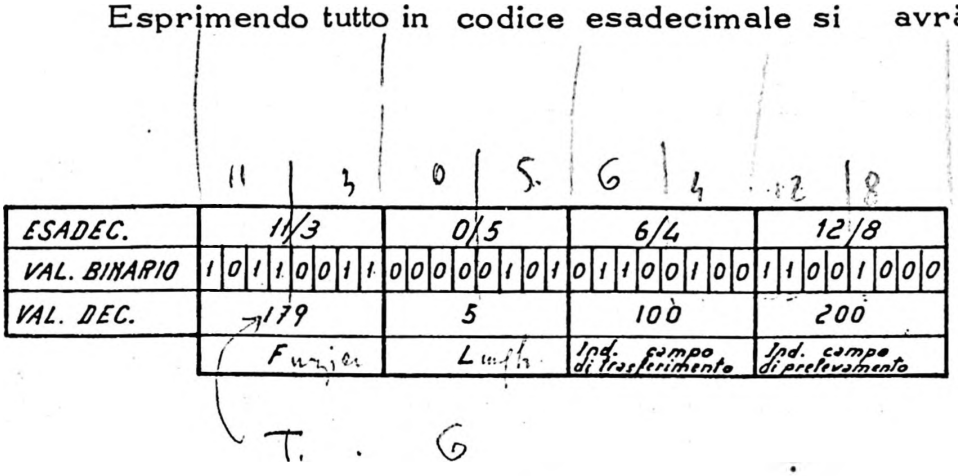
d) - 14 essendo $2^{14} = 16384$.

4.7. INDIRIZZI PER MEMORIA A 16384 POSIZIONI

Si faccia sempre riferimento alla ipotetica memoria a 256 posizioni. In tal caso si potrebbe facilmente esprimere in modo specifico una istruzione di trasferimento.

Il codice di funzione sia T, il numero dei caratteri costituenti il dato sia 6, l'indirizzo di prelievo 200 e quello nel quale si vuole iniziare a porre i dati 100.

Esprimendo tutto in codice esadecimale si avrà:



In tale modo l'istruzione occuperebbe 4 posizioni di memoria.

Questo non é il caso del GE - 115, in quanto la sua capacità minima di memoria é di 4096 posizioni.

Per poter raggiungere valori superiori a 256 bisogna aggiungere dei bit a quelli del carattere che esprimono l'indirizzo.

Ciò é ottenuto esprimendo l'indirizzo non con un solo carattere (otto bit) bensì con due.

Il primo carattere forma i bit aggiuntivi; il secondo carattere interessa integralmente con tutti i suoi 8 bit.

bit aggiunti	15	14	13	12	11	10	09	08
valore decim.			8192	4096	2048	1024	512	256
bit utiliz. nel I° car. dell'indiriz.			05	04	03	02	01	00

Un esempio chiarirà meglio : si debba trasferire un dato di sei caratteri posto nella posizione 2000 alla posizione 3470.

L'istruzione che corrisponde a tale operazione nel GE - 115 sarà espressa in esadecimale come segue :

| | | |

IL PRIMO CARATTERE DELL'INDIRIZZO DEL CAMPO E (PARTE PIU' SIGNIFICATIVA DELL'INDIRIZZO) VIENE INDICATO CON IE^+.

IL SECONDO CARATTERE del campo E (PARTE MENO SIGNIFICATICA DELL'INDIRIZZO) VIENE INDICATO CON IE^-.

ANALOGAMENTE L'INDICAZIONE CORRISPONDENTE PER IL CAMPO A é IA^+, IA^-.

DOMANDE

a) - Con quanti caratteri si esprime un indirizzo ?

Due

b) - Di quanti caratteri differisce l'istruzione reale da quella ideale ? dell'indirizzo

per l'esempio fatto (4+4 cifre + conF + f.)

Per l'esempio fatto 6/10

c) - Quanti bit del primo carattere dell'indirizzo si utilizzano ?

6

d) - Che pesi si attribuiscono ai bit utilizzati del primo carattere dell'indirizzo ?

..08 - 09 - 10 - 11 - 12 - 13

256 - 512 - 1024 - 2048 - 4096 - 8192

e) - Si esprimano in esadecimale e in esadecimale codificato i seguenti indirizzi :

8000, 3225, 4071.

8000 = 1-15 / 4-10 // 1-E / 4-0

3225 = 0-12 / 9-9 // 0-C / 9-9

4071 = 0-15 / 14/7 // 0-F / E-7

RISPOSTE

a) - Con due.

b) - 4. Quella ideale é lunga 10, quella reale é lunga 6.

c) - Un massimo di 6 per ottenere 16384 configurazioni.

d) - (08) 256 - (09) 512 - (10) 1024 - (11) 2048 - (12) 4096 - (13) 8192.

e) - (8000) = 1/15 - 4/0; 1/F - 4/0

(3225) = 0/12 - 9/9; 0/C - 9/9

(4071) = 0/15 - 14/7; 0/F - E/7

LEZIONE 5

Tempo di studio
20 minuti

5. ISTRUZIONI DEL GE-115

5.1. RAGGRUPPAMENTO DELLE ISTRUZIONI

Ogni singola istruzione ha una funzione specifica nello svolgimento del programma.

L'istruzione, interpretata dall'elaboratore, fà eseguire allo stesso solamente l'operazione per la quale é stata progettata.

I tipi di operazione che le istruzioni sono in grado di fare, si possono raggruppare in :

- operazioni di trasferimento
- operazioni logiche
- operazioni aritmetiche
- operazioni di confronto
- operazioni di ricerca
- operazioni di salto
- operazioni esterne

OPERAZIONI DI TRASFERIMENTO

Le istruzioni che appartengono a questo gruppo consentono all'elaboratore di trasferire dei dati da una zona prestabilita ad un'altra di memoria.

OPERAZIONI LOGICHE

Tali operazioni consistono nell'agire in maniera logica sui dati per ottenere delle trasformazioni, permanenti o momentanee, degli stessi.

OPERAZIONI ARITMETICHE

Esse permettono di operare aritmeticamente su due numeri contenuti in due zone diverse di memoria.

OPERAZIONI DI CONFRONTO

Confrontano tutti gli otto bit o soltanto i quattro meno significativi dei caratteri di due dati posti in zone diverse di memoria.

OPERAZIONI DI RICERCA

Ricercano un carattere prestabilito, tra i dati contenuti in un campo di memoria.

OPERAZIONI DI SALTO

Permettono, al verificarsi o non verificarsi di determinate condizioni interne od esterne all'elaboratore, di interrompere o modificare la normale sequenza delle istruzioni.

Il programma procede saltandone alcune o ritornando ad una parte precedentemente svolta per ripeterla.

OPERAZIONI ESTERNE

Esse si riferiscono allo scambio di dati, all'esame delle condizioni, all'invio dei comandi che l'elaboratore deve effettuare con le macchine elettromeccaniche ad esso allacciate.

DOMANDE

a) - Quale operazione definita può fare un'istruzione ?
L'istruzione fa quello è stata profetizzata

b) - Quanti scopi ha un'istruzione ? Uno
Scopo di eseguire quella data istruzione

c) - In quanti tipi di operazione si possono raggruppare le istruzioni ?
7

d) - Quali sono le operazioni ?
trasferimento
logiche
aritmetiche
confronto
ricerca
salto
Esterne

RISPOSTE

a) - Quella per la quale é stata progettata.

b) - Uno e uno solo.

c) - Sette.

d) - Operazioni di trasferimento, logiche, aritmetiche, di confronto, di ricerca, di salto, esterne.

5.2. GRUPPI DI ISTRUZIONI

Le istruzioni del GE - 115 sono di due tipi :

- istruzioni che interessano solo organi interni dell'U.C. (istruzioni interne)
- istruzioni che interessano anche organi esterni all'U.C. (istruzioni esterne).

Delle istruzioni esterne si parlerà in seguito.

Si inizia ora la presentazione di quelle interne.

Queste ultime formano 5 gruppi distinti.

Ogni gruppo comprende istruzioni aventi caratteristiche comuni.

I gruppi vengono distinti con le seguenti sigle :

Gruppo A1

Gruppo A2

Gruppo B3

Gruppo C4

Gruppo D4

Si inizia ora l'analisi delle caratteristiche comuni alle istruzioni appartenenti ad ogni singolo gruppo.

5.3. CARATTERISTICHE COMUNI ALLE ISTRUZIONI DEL GRUPPO A2

Per semplicità d'esposizione si inizia a considerare le caratteristiche delle istruzioni del gruppo A2.

NUMERO DI CARATTERI DELL'ISTRUZIONE : 6

1°	2°	3°	4°	5°	6°
F	L	IE^+	IE^-	IA^+	IA^-
F	L	IE		IA	

FASE DI ESECUZIONE

I dati o operandi vengono interessati solo per indirizzi crescenti.

Cioé se a partire dall'indirizzo 1000 di memoria si trova un operando, si inizia ad elaborarlo incominciando dall'indirizzo 1000 e procedendo per posizioni di memoria crescenti (1000 - 1001 - 1002 ecc.).

INDIRIZZI INIZIALI DEI CAMPI

Poiché durante la fase di esecuzione l'elaboratore opera per indirizzi crescenti, IE ed IA devono definire la posizione più significativa dei rispettivi operandi.

Per posizione più significativa s'intende l'indirizzo del rango decimale più elevato del dato sul quale si deve operare (per esempio il 4 del numero 46.723).

LUNGHEZZA DELL'OPERANDO
(riferita al campo E)

La lunghezza di un operando può essere al massimo di 256 caratteri. Infatti la lunghezza si esprime

con un solo carattere.

Poiché ogni carattere é composto da otto bit le combinazioni possibili (caratteri diversi) possono essere al massimo $2^8 = 256$.

Con lunghezza 0 si elabora 1 carattere.

La lunghezza di un operando é facilmente calcolabile in quanto corrisponde alla differenza tra gli indirizzi estremi dell'operando del campo.

Ad esempio se il dato 46.723 é posto nelle posizioni da 1000 a 1004 di memoria la sua lunghezza é determinata da 1004 - 1000 =4 (numero dei caratteri che formano il dato=5 per cui : L = 5 - 1 = 4).

Nelle lezioni seguenti quando si fa riferimento al valore del carattere di lunghezza L si deve sempre considerare che si opera su un numero di caratteri corrispondenti a L + 1.

RISULTATO

L'eventuale risultato delle operazioni si trova sempre nel campo E; per cui l'operando, che precedentemente si trovava in tale campo, viene sostituito con il risultato dell'operazione.

Il campo A rimane inalterato.

5.4. ESEMPIO DI INTERPRETAZIONE DI UNA ISTRUZIONE DI GRUPPO A2

Una istruzione appartenente a tale gruppo é la CMC.

Con tale simbolo si indica l'istruzione che esegue un operazione di confronto.

Il confronto viene eseguito tra un numero di caratteri, espressi dalla lunghezza L, appartenenti al campo E e quelli corrispondenti appartenenti al campo A.

Il contenuto dei due campi in questo caso rimane immutato.

L'istruzione quindi dovrà indicare all'elaboratore con :

F = D/5 → che é un'istruzione di confronto (CMC)

L = X — che X é il numero dei caratteri del campo E che interessa confrontare con altrettanti caratteri del campo A

$IE = IE^+, IE^-$ — che IE^+ ed IE^- esprimono l'indirizzo iniziale del campo E

$IA = IA^+, IA^-$ — che IA^+ ed IA^- esprimono l'indirizzo iniziale del campo A.

Al termine, l'operazione di confronto segnala all'elaboratore, tramite appositi organi che in seguito verranno presentati, se il contenuto del campo E é uguale o diverso da quello del campo A. (E = A, E ≠ A).

Tale segnalazione potrà quindi essere utilizzata da una successiva istruzione atta a considerare l'indicazione contenuta in tali organi.

IN DEFINITIVA L'OPERAZIONE DI CONFRONTO HA LO SCOPO DI CONFRONTARE UNO AD UNO, UN CERTO NUMERO DI CARATTERI APPARTENENTI AI DUE CAMPI E SEGNALARE AD ALCUNI ORGANI IL RISULTATO.

Il modo di analizzare un'istruzione rimane sempre lo stesso per tutte le istruzioni appartenenti al medesimo gruppo; é importante tenere sempre presente i seguenti punti :

- l'ordine con il quale sono interessati gli operandi (indirizzi crescenti)
- il numero dei caratteri da elaborare (lunghezza L)
- gli indirizzi iniziali dei campi (IE,IA)
- l'eventuale risultato si sostituisce all'operando posto nel campo E.

DOMANDE

Si supponga di avere la seguente istruzione posta all'indirizzo 0/0 di memoria :

MVC		163		176	
D/2	0/5	0/0	A/3	0/0	B/0
F	L	IE+	IE-	IA +	IA-

Tenendo presente che il D/2 corrisponde all'istruzione MVC (trasferimento) rispondere alle seguenti domande :

a) - Da dove inizia l'elaboratore a prelevare il dato ?

nel campo A mezzo 176 (B0)

b) - In quale campo si pone il dato trasferito ?

nel campo F mezzo 163

c) - Quanti caratteri vengono trasferiti ?

X(0-5) = 6

d) - Qual'é l'indirizzo finale del campo ove viene trasferito il dato ?

163 + 6 = 168 = (A3 + 5) = A8

RISPOSTE

a) - L'elaboratore inizia a prelevare il dato dal campo A quindi dall'indirizzo 0/0 - B/0 (176).

b) - Il campo in cui si pone il dato é il campo E.

c) - Poiché la lunghezza é 0/5 i caratteri trasferiti sono 6.

d) - L'indirizzo finale del campo in cui viene posto il dato (campo E) é 0/0 - A/8 (168). Infatti essendo 6 i caratteri da trasferire ed iniziando il trasferimento dall'indirizzo 0/0 - A/3 e poiché l'elaboratore procede per indirizzi crescenti, l'indirizzo finale sarà 0/0 - A/3, 0/0 - A/4 ..... 0/0 - A/8.

LEZIONE 6

Tempo di studio
1 ora e 30 minuti

6. ISTRUZIONI DEL GRUPPO A2

6.1. Si riporta ora l'elenco completo delle istruzioni appartenenti al gruppo A2.

Per ogni istruzione vengono indicate le caratteristiche essenziali : l'operazione per la quale un'istruzione é stata ideata, i particolari più interessati che si ottengono a fine istruzione ecc.

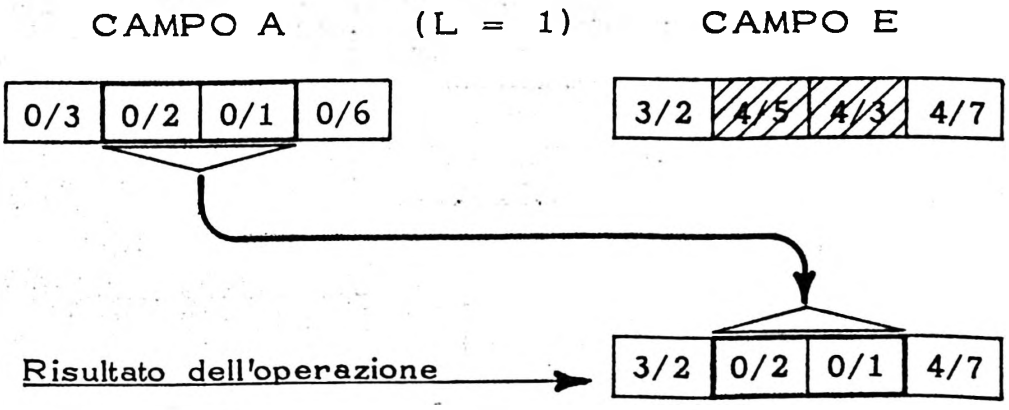
6.2. TRASFERISCE IN ALFANUMERICO

MVC (Move Complete Octets).

D/2

Trasferisce il contenuto del campo A in campo E per lunghezza L.

Vengono trasferiti tutti gli 8 bit di ogni carattere. Il contenuto del campo A rimane immutato.



6.3. CONFRONTO ALFANUMERICO

CMC (Compare Complete Octets).

D/5

Confronto fra il contenuto del campo E e il contenuto del campo A per lunghezza L. riferita ad entrambi i campi.

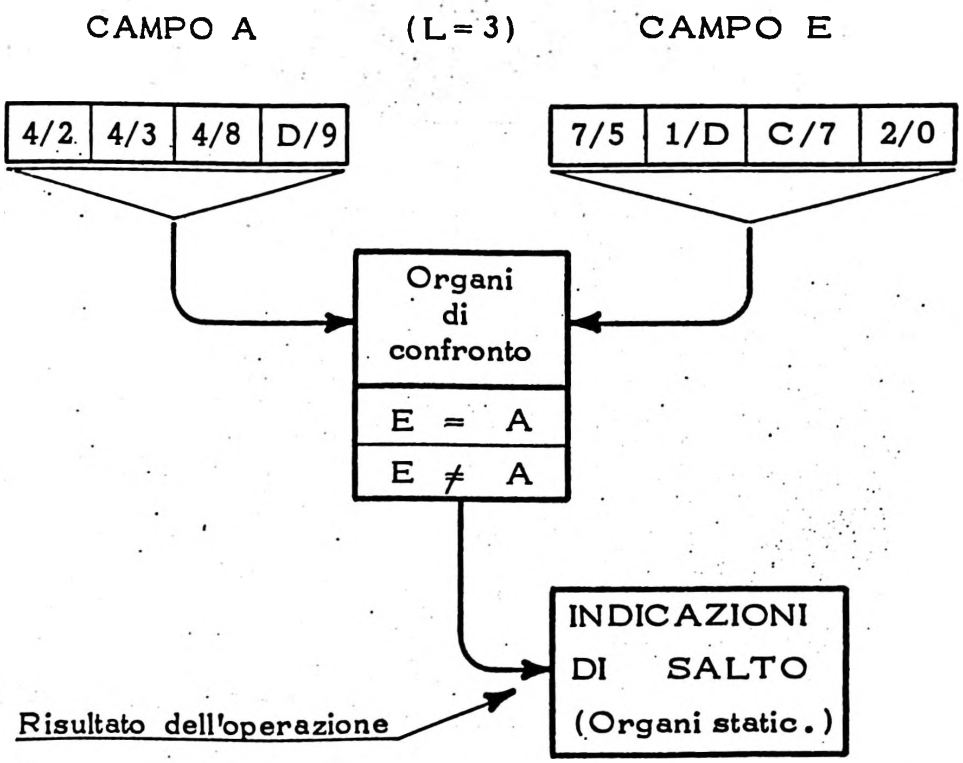
Sono confrontati tutti i bit (8) di ogni carattere.

Il contenuto dei campi rimane immutato.

Si dispongono gli organi che staticizzano il risultato del confronto utilizzabile mediante le istruzioni di salto.

Il risultato del confronto può essere :

$E = A$, $E \neq A$



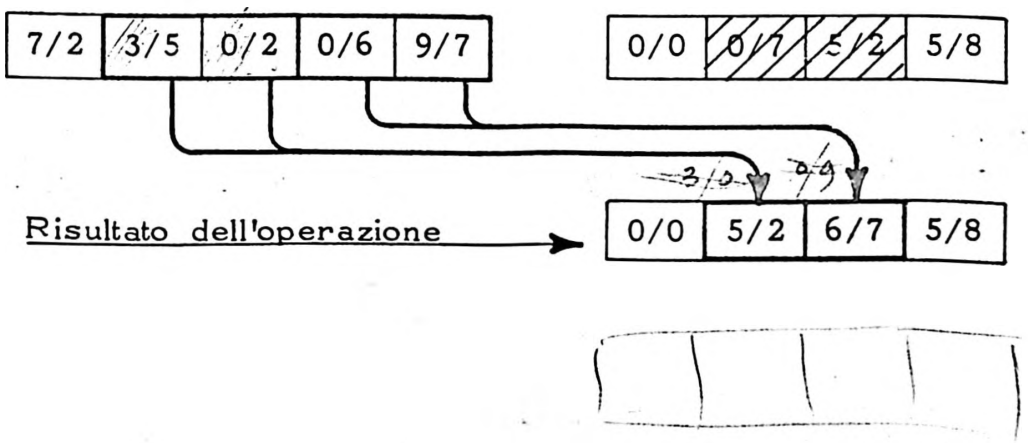
6.4. IMPACCA

PK
D/A

(Pack Right Quartets into Octets).

Impaccamento di ogni coppia di caratteri del campo A in un unico carattere che viene posto in campo E.

I 4 bit meno significativi del primo e del secondo carattere del campo A compongono rispettivamente i 4 bit più ed i 4 bit meno significativi del carattere E. Il campo A resta intatto. La lunghezza L si riferisce al campo E.



6.5. DISIMPACCA

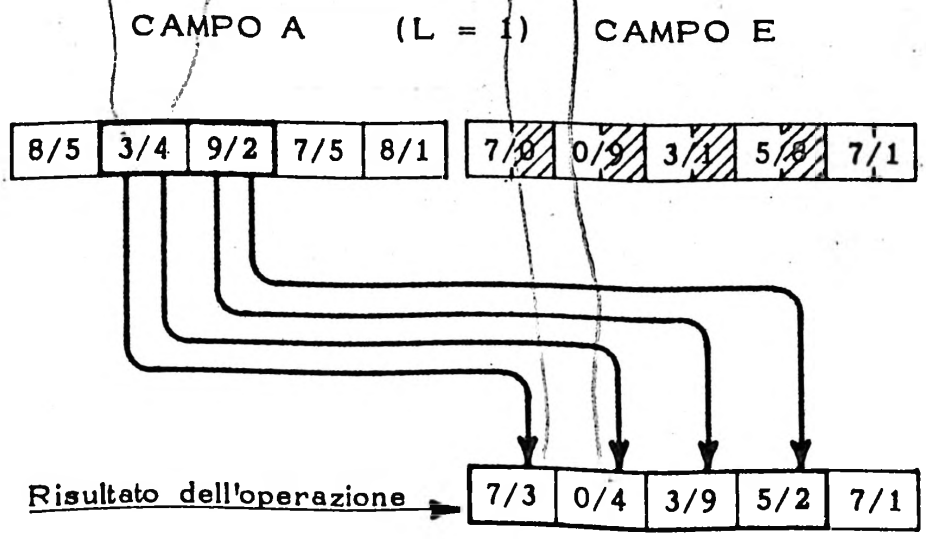
UPK
D/8

(Unpack Octets into Right Quartets).

Disimpaccamento di ogni carattere del campo A in due distinti caratteri che vengono disposti di seguito in campo E.

I 4 bit più e i 4 bit meno significativi del carattere del campo A compongono rispettivamente i 4 bit meno significativi del primo e del secondo carattere del campo E. I 4 bit più significativi di ciascun carattere del campo E restano immutati.

La lunghezza L (contrariamente a quanto si verifica per le istruzioni di questo gruppo) é riferita al campo A.



6.6. OR

OR su tutti i bit.

OC

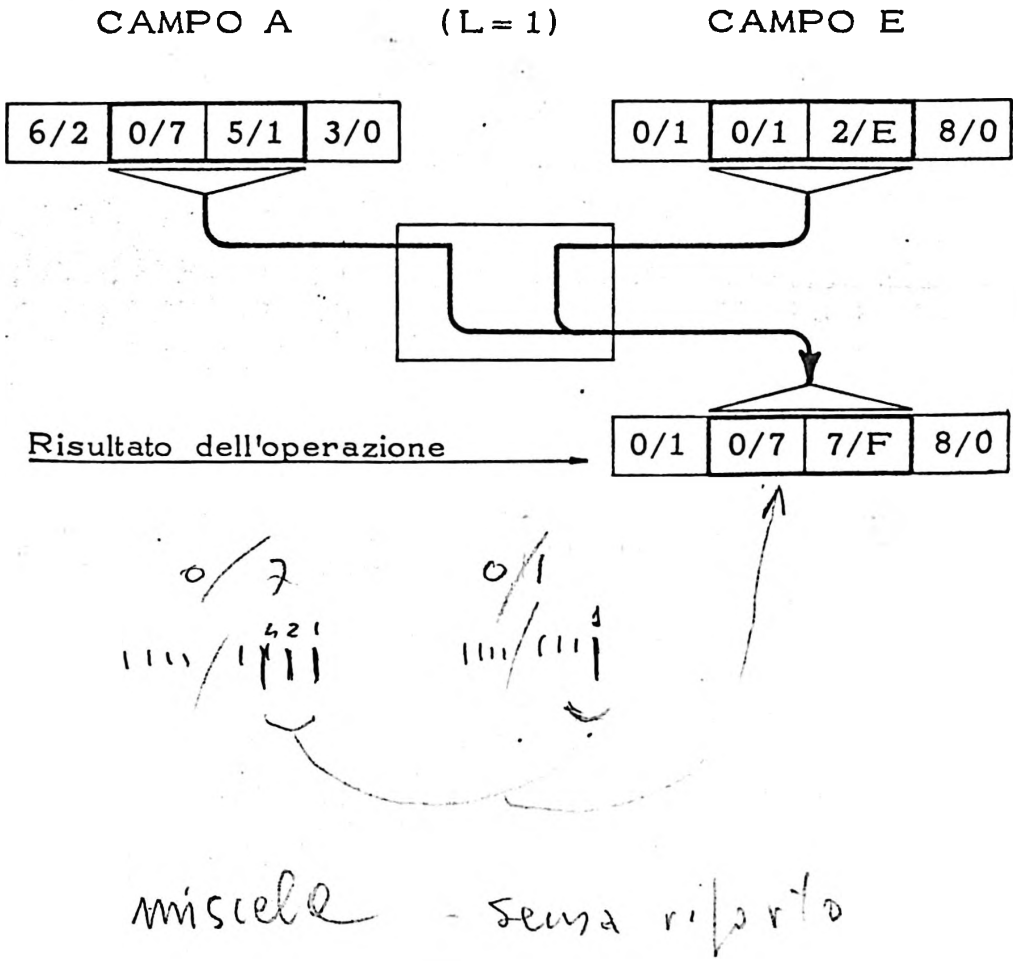
D/6

(OR on Complete Octets).

Or logico di ogni carattere del campo A con ogni carattere corrispondente del campo E per una lunghezza L riferita ad entrambi i campi.

L'OR risultante fra gli otto bit di ogni carattere viene posto in campo E.

Il campo A rimane immutato.



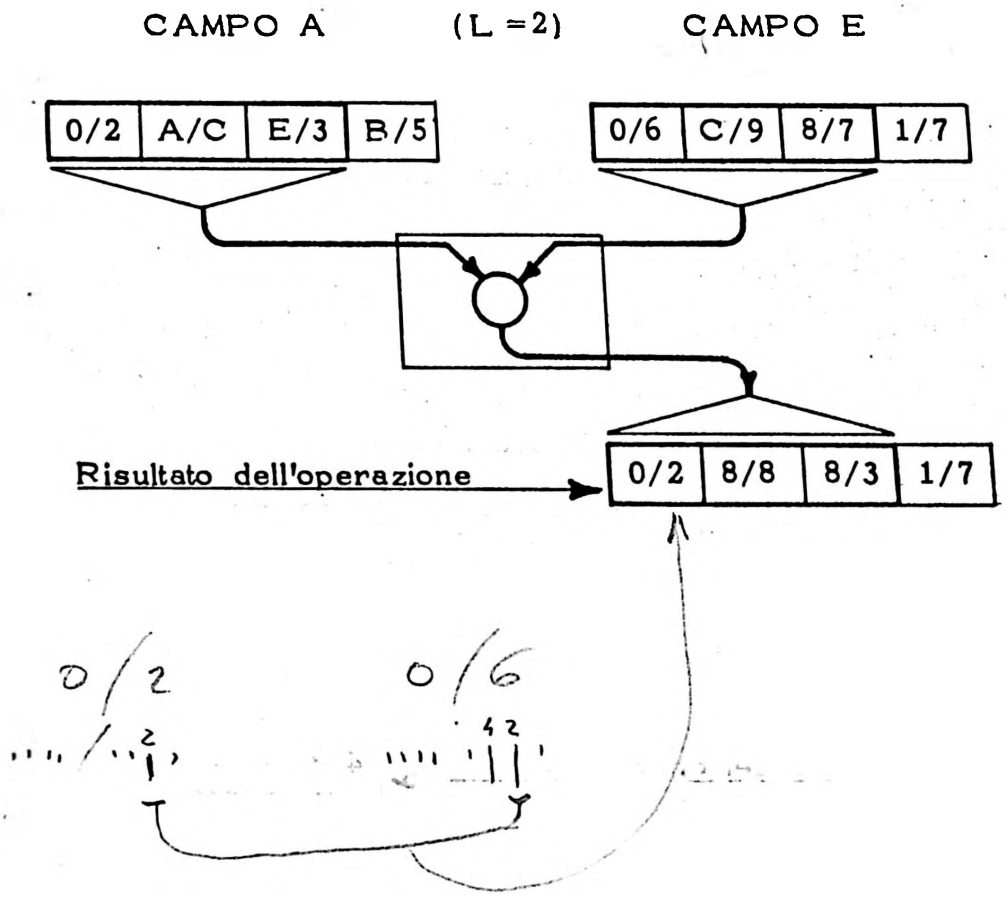
miscela - senza riporto

6.7. AND

NC
D/4

(And on Complete Octets).

And logico, eseguito con le stesse modalità dell'OR logico.



Somme senza riport. di tutti i bit.

6.8. OR ESCLUSIVO

$A\bar{B} + B\bar{A}$

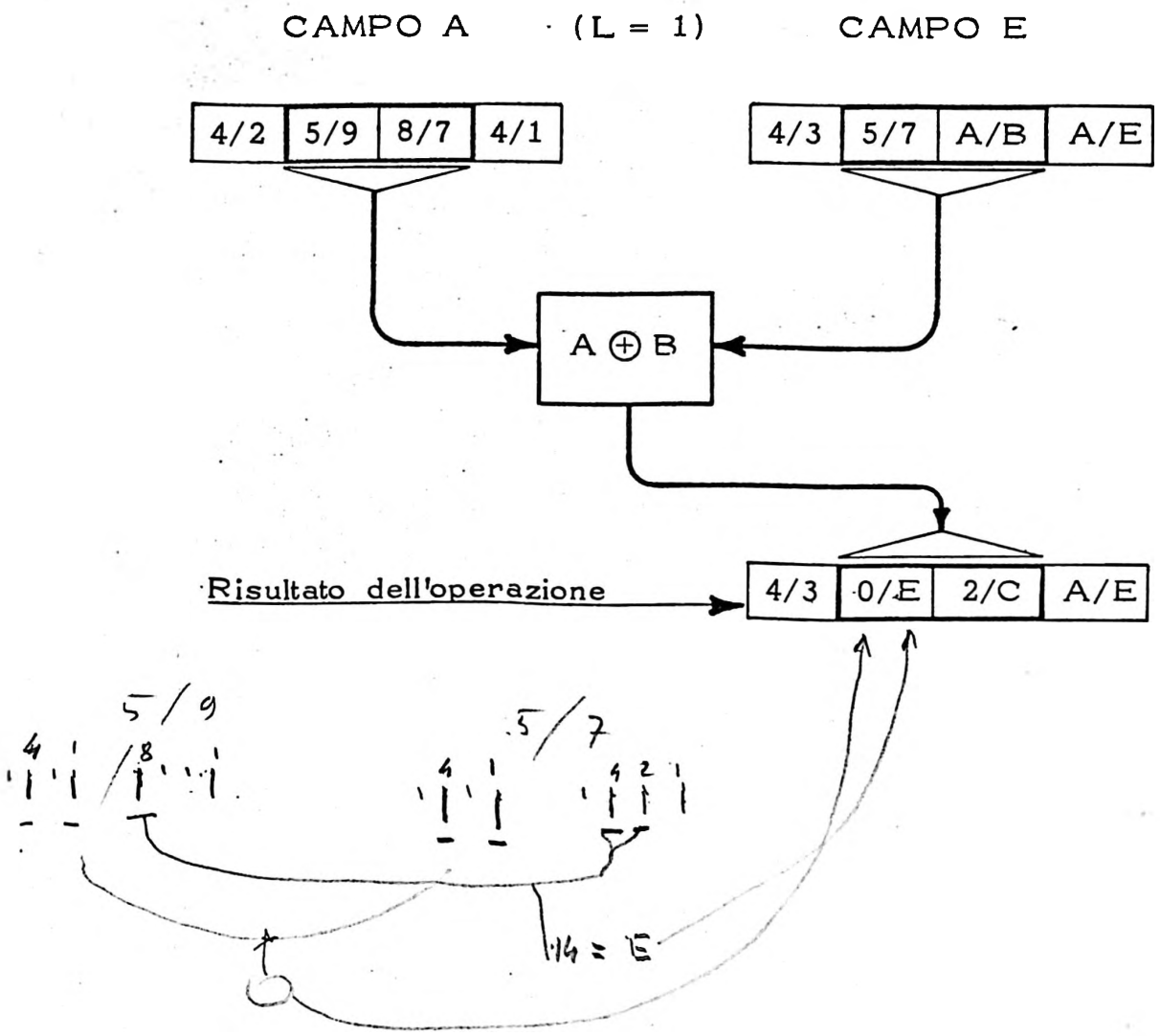
XC
D/7

(Exclusive Or on Complete Octets).

L'Or esclusivo si può intendere come somma senza trasmissione di riporto tra i bit del campo A con quella di ogni carattere del campo E. Viene fornita una condizione di salto in base al risultato.

(= a zero o ≠ da zero).

Il campo A rimane immutato.



6.9. RICERCA A DESTRA

SR
D/9

(Search to the Right).

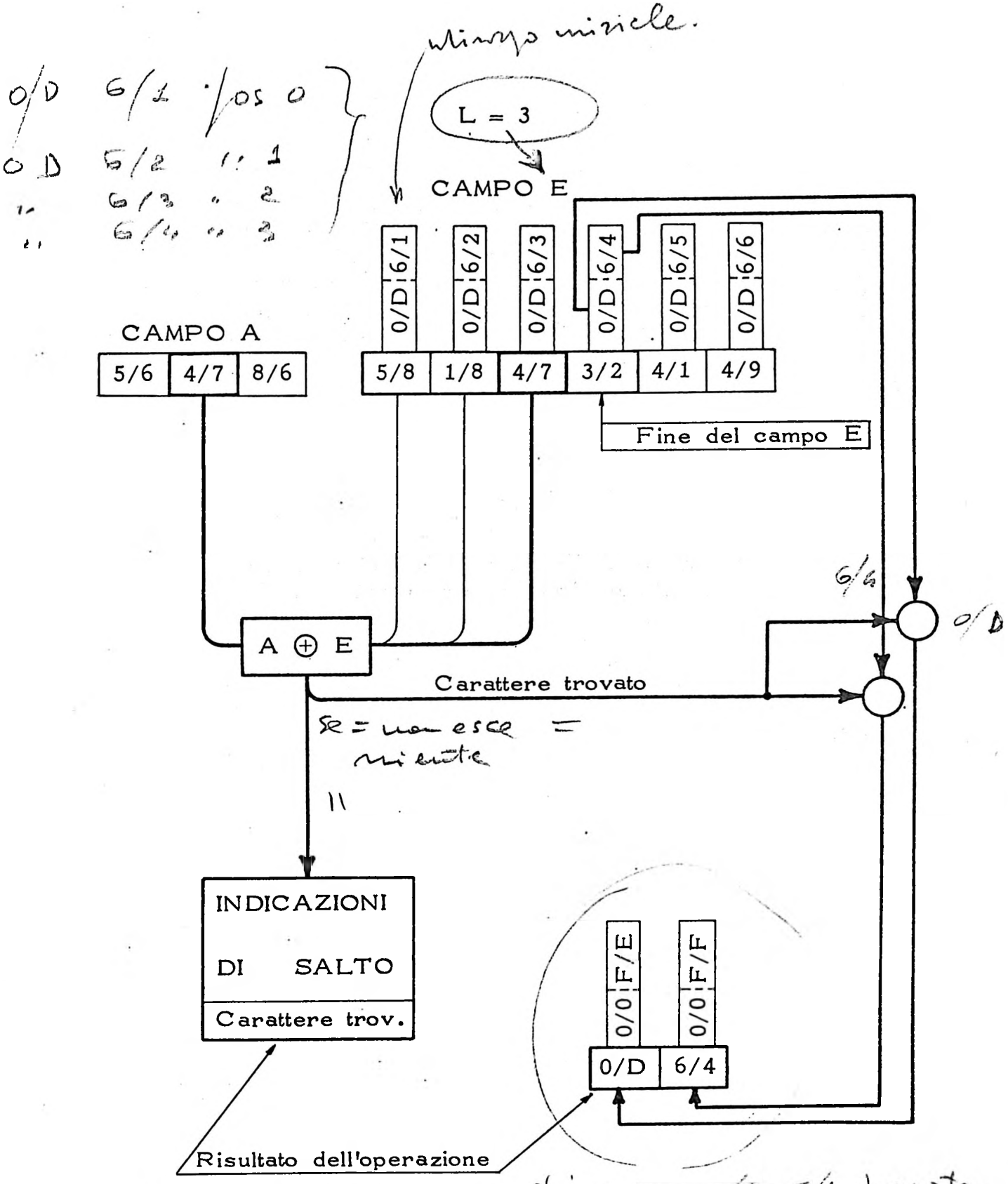
Ricerca per indirizzi crescenti, di un carattere uguale a quello posto all'indirizzo IA, fra i caratteri contenuti nel campo E per una lunghezza L riferita al campo E.

L'operazione può presentare due casi distinti :

1° caso :

il carattere ricercato viene trovato.

In questo caso l'operazione termina con il ritrovamento del carattere e si ha come risultato, la registrazione nelle posizioni di memoria F/E - F/F (254 - 255) dell' indirizzo successivo a quello nel quale si trova il carattere ricercato.


0/D 6/1 pos 0
0 D 6/2 " 1
" 6/3 " 2
" 6/4 " 3
tempo iniziale.
L = 3
CAMPO E
CAMPO A
5/6
4/7
8/6
0/D 6/1
0/D 6/2
0/D 6/3
0/D 6/4
0/D 6/5
0/D 6/6
5/8
1/8
4/7
3/2
4/1
4/9
Fine del campo E
6/4
0/D
A ⊕ E
Carattere trovato
se = non esce = rimane
INDICAZIONI
DI SALTO
Carattere trov.
0/0 F/E
0/0 F/F
0/D
6/4
Risultato dell'operazione
(posiz. successive a quella contenente il carattere trovato)
il indirizzo 0/D - 6/4 trovato,
lo scrive nella posizione di memoria
0/0 - FE e 0/0 - FF
6-11
prefissate
(Fisse ?)

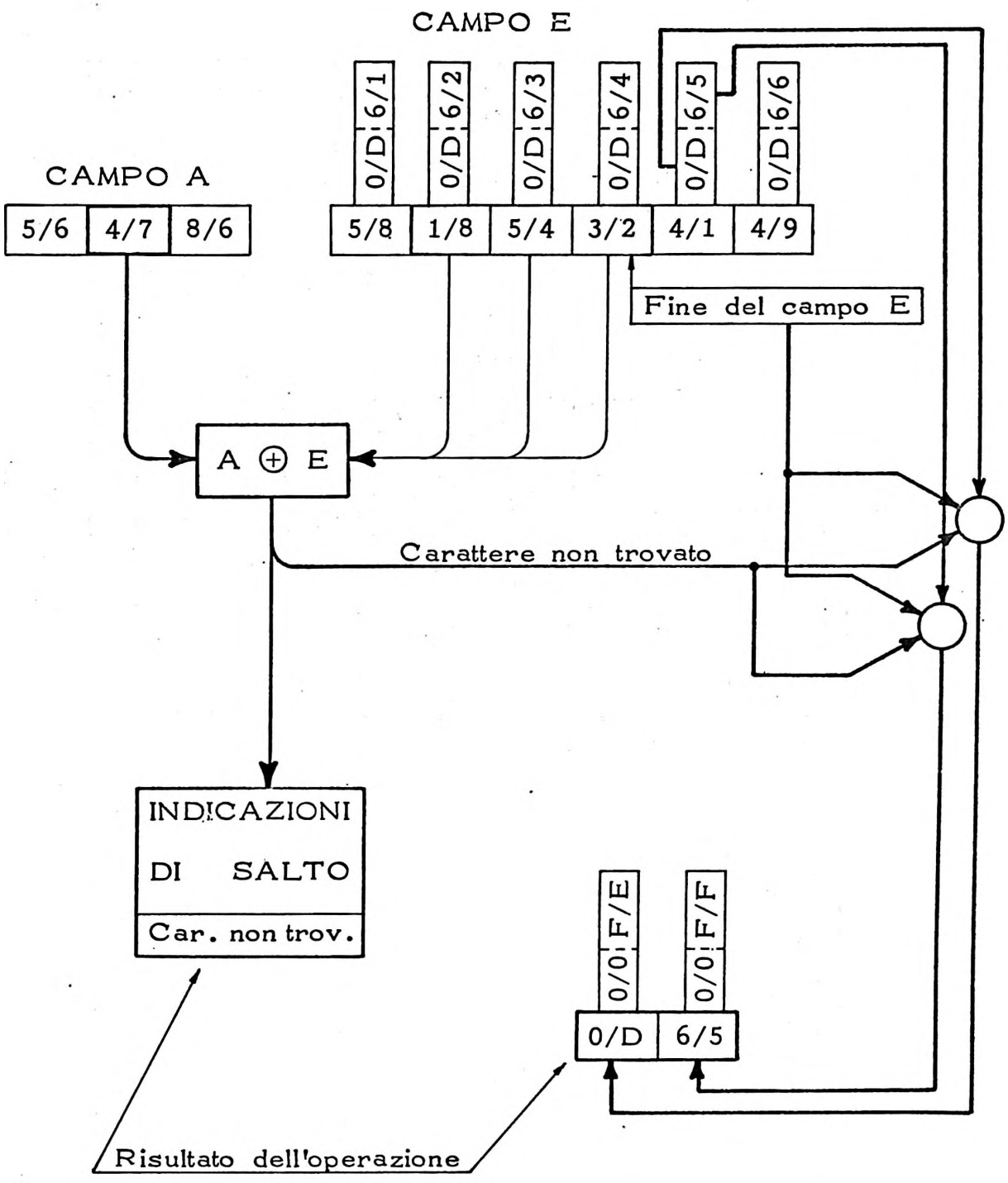
2° caso :

il carattere ricercato non viene trovato.

In questo caso l'operazione termina dopo aver esaminato tutto il campo E (fine su lunghezza) e fornisce come risultato la registrazione nelle posizioni di memo - ria F/E - F/F (254 - 255) l'indirizzo suc cessivo a quello della fine del campo E.

In entrambi i casi vengono fornite delle condizioni di salto tramite le quali é possibile, da programma, stabilire se la istruzione é terminata, con il ritrovamento o no del carattere ricercato.

L = 2


CAMPO E
0/D 6/1
0/D 6/2
0/D 6/3
0/D 6/4
0/D 6/5
0/D 6/6
CAMPO A
5/6
4/7
8/6
5/8
1/8
5/4
3/2
4/1
4/9
Fine del campo E
A ⊕ E
Carattere non trovato
INDICAZIONI
DI SALTO
Car. non trov.
0/0 F/E
0/0 F/F
0/D
6/5
Risultato dell'operazione

6.10. RICERCA A SINISTRA

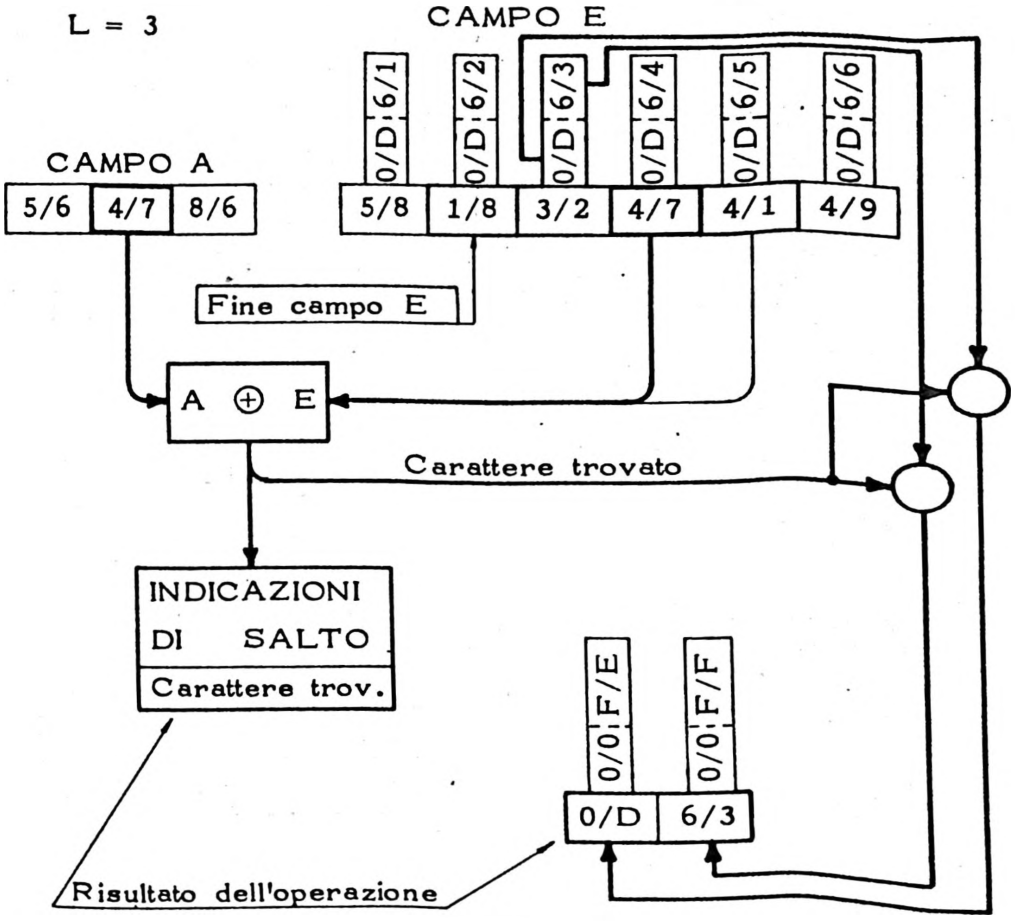
SL
D/B

(Search to the Left).

Ricerca indietro di un carattere, eseguita con le stesse modalità della SR, con la eccezione che viene indirizzato il carattere meno significativo di E, e l'operazione procede per indirizzi decrescenti.

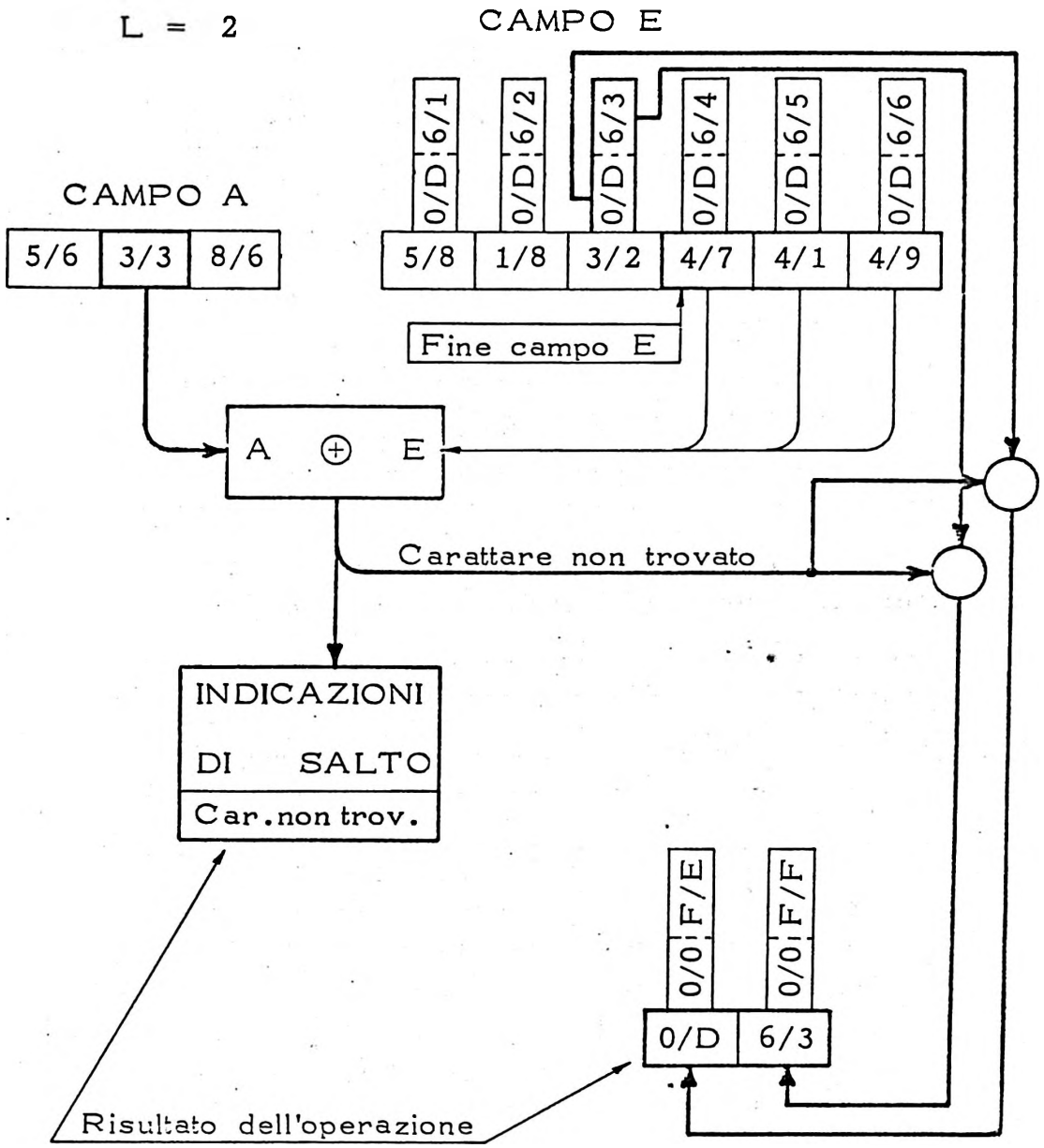
1° caso :

il carattere é stato trovato.



2° caso :

il carattere non é stato trovato.



DOMANDE C / A E

a) - Analizzare la seguente serie di caratteri :

D/C 0/5 0/1 0/1 2/0 0/2

e rispondere alle seguenti domande :

1) Si può considerare una istruzione del gruppo A2 ? → → OR logico

2) Se é un'istruzione quale operazione svolge ?

3) Quanti caratteri vengono elaborati ? — 6

4) Qual'é l'indirizzo iniziale, espresso in decimale, del campo E ? 257 ~~34~~

5) Qual'é l'indirizzo iniziale espresso in decimale del campo A ? 8194 ~~DX~~

.....
.....

b) - L'elaboratore procede all'analisi di tutte le istruzioni del gruppo A2 per indirizzi crescenti ?

No (ovvero Si escluse SL)
.....

c) - Che utilità presenta l'operazione d'impaccamento ?

diminuire il N° di posiz. occupate ma
.... conf. ... lavorarli

d) - Durante l'operazione di disimpaccamento quante posizioni del campo E sono interessate rispetto a quelle del campo A ?

1 solo

e) - Quando nelle operazioni di ricerca (avanti o indietro) si trova il carattere ricercato quale altra operazione esegue l'elaboratore prima di terminare la istruzione ?

scrive l'indirizzo successivo alla posiz trovata / o l'indirizzo successivo
al campo E se non è stato trovato il carattere
cerca le istruzioni per il salto = 0 ≠

f) - A quale operazione binaria può essere paragonato l'OR esclusivo ?

miscela dei bit. diversi (Addizione senza riport.)

g) - Si supponga di avere nella posizione di memoria 1050 il carattere 0/3. Si vuole ottenere nella posizione 3000 il carattere F/F sapendo che in tale posizione c'é già il carattere F/D quale operazione logica é necessario eseguire ?
Riportare nelle caselle sottostanti l'istruzione completa

O.C D/C	0	0/B	B/8	0/4	1/A
F	L	IE		IA	

h) - Se nella posizione di memoria 3200 c'é il carattere A/8 e nella posizione 7025 il carattere 7/3 quale sarà l'istruzione completa necessaria ad ottenere nella posizione 3200 il carattere 2/0 ?

(AND) Dh ecc.

i) - A quale campo é riferita la lunghezza nelle istruzioni del gruppo A2 ?

Al campo F

Nelle UPK è riferito al campo A

RISPOSTE

a) - 1) Si, perché il codice D/C corrisponde all'istruzione OC.

2) Svolge un'operazione di OR logico.

3) Essendo la lunghezza espressa 0/5 si elaborano 6 caratteri.

4) 257.

5) 8194.

b) - Tutte le istruzioni, ad eccezione della SL, vengono eseguite per indirizzi crescenti.

c) - Ha l'utilità di poter mettere in una sola posizione di memoria le parti meno significative di due caratteri. Ciò significa che, essendo la parte meno significativa di un carattere la parte numerica decimale, se si lavorasse con soli numeri le posizioni di memoria potrebbero considerarsi doppie rispetto a quelle occupate con caratteri alfabetici.

d) - Il doppio, poiché le parti più e meno significative di ogni carattere del campo A vengono portate nelle parti meno significative di due caratteri del campo E.

e) - Alla fine della SL e SR l'elaboratore esegue sempre l'immagazzinamento dell'indirizzo immediatamente seguente quello in cui si trova il carattere cercato, nelle posizioni di memoria 255-254. Se il carattere non viene trovato l'istruzione termina su lunghezza e si immagazzina sempre l'indirizzo seguente all'ultima posizione interessata.

f) - Ad un operazione binaria di somma o di sottrazione senza trasmissione dei riporti.

g) - L'istruzione é una OC (OR logico) che dovrà essere impostata nel seguente modo :

D/6	0/0	0/B	B/8	0/4	1/A
F	L	IE		IA	

h) - L'istruzione é una NC (AND logico) che dovrà essere scritta nel modo seguente :

D/4	0/0	0/C	8/0	1/B	7/1
F	L	IE		IA	

i) - La lunghezza é sempre riferita al campo E, ad eccezione della UPK nella quale la lunghezza é riferita al campo A.

6.11. Al gruppo A2 appartengono anche le istruzioni EDT (Edizione stampa) e TR (trascodifica).

Queste istruzioni sono presentate successivamente e cioé la TR nell'ultima lezione del presente volume mentre la EDT é nel volume "U.C. vol. 3°".

LEZIONE 7

Tempo di studio
20 minuti

7. CARATTERISTICHE COMUNI ALLE ISTRUZIONI DEL GRUPPO A1

7.1. Le istruzioni appartenenti a tale gruppo sono composte da sei caratteri come quelle del gruppo A2.

LA PARTICOLARITA' PIU' INTERESSANTE CHE DIFFERENZIA LE ISTRUZIONI DEL GRUPPO A1 DA QUELLE DEL GRUPPO A2 RIGUARDA IL MODO IN CUI L'ELABORATORE INTERPRETA IL CARATTERE CHE INDICA LA LUNGHEZZA DEI DATI DA ELABORARE.

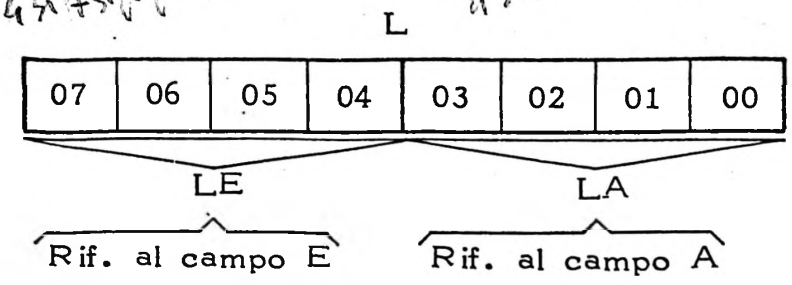
Nel gruppo precedente la lunghezza espressa nel l'istruzione, relativa ai dati da elaborare, esprime con tutti i suoi bit il numero massimo di 256 caratteri sui quali si può operare.

Nel gruppo A1 lo stesso carattere L viene interpretato dall'elaboratore in modo differente.

Cioé ai bit del carattere che esprime la lunghezza vengono attribuite due funzioni diverse :

- i quattro bit più significativi determinano la lunghezza dell'operando del campo E
- i quattro bit meno significativi determinano la lunghezza dell'operandi del campo A.

La lunghezza massima esprimibile in questa istruzione, sia per il campo E che per il campo A risulta di $2^4 = 16$ caratteri.



Le caratteristiche comuni di queste istruzioni sono :

ORDINE DI ESECUZIONE

Le istruzioni elaborano i dati per indirizzi decrescenti.

Se, per esempio, s'inizia ad operare dall'indirizzo 1000 i successivi caratteri verranno estratti da posizioni di indirizzo inferiore (999 - 998 ecc.).

INDIRIZZI INIZIALI DEI CAMPI

Gli indirizzi espressi nell'istruzione devono definire le posizioni meno significative dei due operandi (per esempio, l'indirizzo iniziale dell'operando 4792 é quello del carattere 2).

LUNGHEZZE DEGLI OPERANDI

Sono definite come differenze tra gli indirizzi estremi dei campi (numero dei caratteri da elaborare meno uno). Il numero massimo di caratteri sui quali si può operare per ogni campo é 16. (L = F)

RISULTATO

L'eventuale risultato delle operazioni si trova sempre nel campo E.

TERMINE DELLA OPERAZIONE

L'operazione termina quando é stato elaborato l'ultimo carattere del campo E. Se LA > LE i caratteri eccedenti il campo A non vengono interessati. Se LA< LE i caratteri mancanti a LA vengono rimpiazzati con zeri fittizi. Se LA= LE vengono interessati tutti i caratteri di entrambi i campi.

DOMANDE

a) - Che differenza esiste tra le istruzioni del gruppo A1 e quelle del gruppo A2 nella interpretazione del codice L ?

(16 car) (max 15 → Nel gruppo A1 i 4 bit + significativi si riferisca al campo F
(256 car) (Max 255 → Nel gruppo A2 mentre i 4 bit + significativi si riferiscono al campo A

b) - Di quanti caratteri sono composte le istruzioni del gruppo A1 ?

6

c) - Come procede lo svolgimento delle istruzioni del gruppo A1 rispetto a quelle del gruppo A2 ?

è ritroso ma D è simultaneo in A2 = crescente

d) - Qual'é la lunghezza massima di un operando di una istruzione del gruppo A1 e qual'é quella di un operando di una istruzione del gruppo A2 ?

A1 = max 16
A2 = Max 256

e) - Le istruzioni del gruppo A1 e quelle del gruppo A2 sono istruzioni di tipo interno od esterno ?

interno

f) - In riferimento alla fine della operazione qual'é la lunghezza che viene presa in considerazione ?

nel campo E

g) - Si abbiano tre istruzioni del gruppo A1 i cui caratteri L hanno i seguenti valori :

ISTRUZIONE	L	
	LE	LA
1	4	7
2	B	6
3	F	F

Qual'é il numero di caratteri elaborati per ciascun campo di ogni istruzione ?

Cp E ..... 5 / 12 / 16 .....
.....

Cp A 5 / 7 / 16

RISPOSTE

a) - Nelle istruzioni del gruppo A1, i bit che formano il carattere che esprimono la lunghezza dell'operando, sono suddivisi in due parti uguali. I quattro bit più significativi si riferiscono al campo E, mentre i quattro meno significativi si riferiscono al campo A, nelle istruzioni del gruppo A2 la lunghezza é riferita ad un solo campo e può esprimere un massimo di 256 caratteri.

b) - Le istruzioni del gruppo A1 sono composte sempre di sei caratteri.

c) - Le istruzioni del gruppo A1 si svolgono per indirizzi decrescenti, quelle del gruppo A2 per indirizzi crescenti(tranne la RCI). SL ricerca a sinistra

d) - Nelle istruzioni del gruppo A1, la lunghezza massima dei campi é $2^4 = 16$, nella A2 di $2^8 = 256$.

e) - Ambedue i gruppi appartengono al tipo di istruzioni interne.

f) - LE, cioé la lunghezza relativa al campo E.

g) -

Istruzione	Caratteri elaborati		Note
	Campo E	Campo A	
1	5	5	Si trascurano 3 caratteri del campo A
2	12	7	Si aggiungono 5 zeri fittizi al campo A
3	16	16	

LEZIONE 8

Tempo di studio
20 minuti

8. INTERPRETAZIONE DI UNA ISTRUZIONE DEL GRUPPO A1

8.1. Fa parte di questo gruppo l'istruzione AD, addizione decimale, così composta :

F	LE	LA	IE^+	IE^-	IA^+	IA^-
F	L		IE		IA	

Si avrà allora con :

F = F/A	l'istruzione AD
L = LE/LA	il numero dei caratteri da elaborare del campo E e del campo A
IE = (IE^+ IE^-)	l'indirizzo iniziale del campo E
IA = (IA^+ IA^-)	l'indirizzo iniziale del campo A.

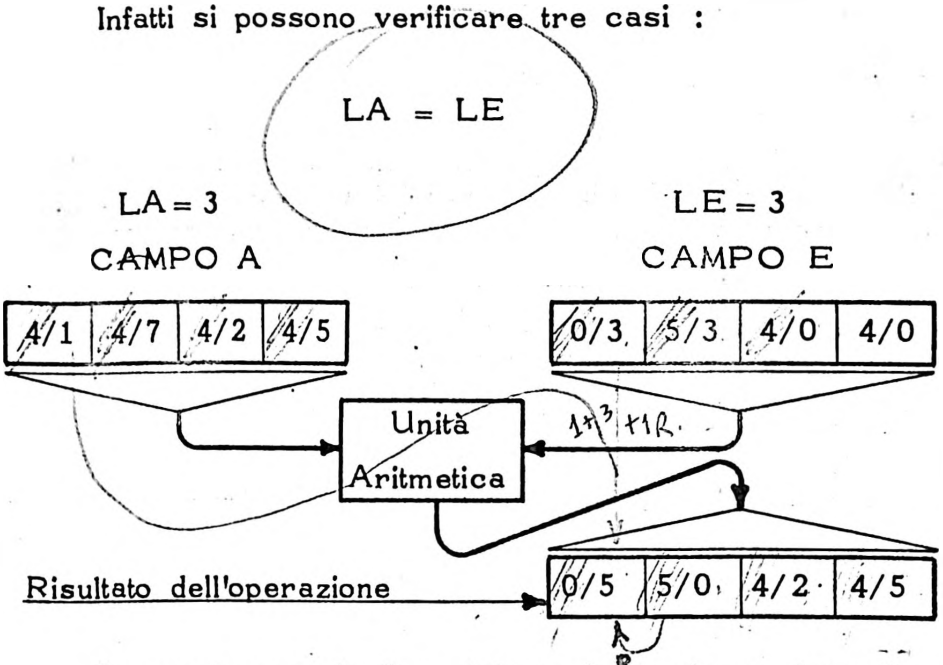
8.2. L'elaboratore opera facendo la somma decimale tra il contenuto del campo A e quello del campo E.

Si opera solo sulla parte numerica del carattere (bit meno significativi 03 - 02 - 01 - 00).

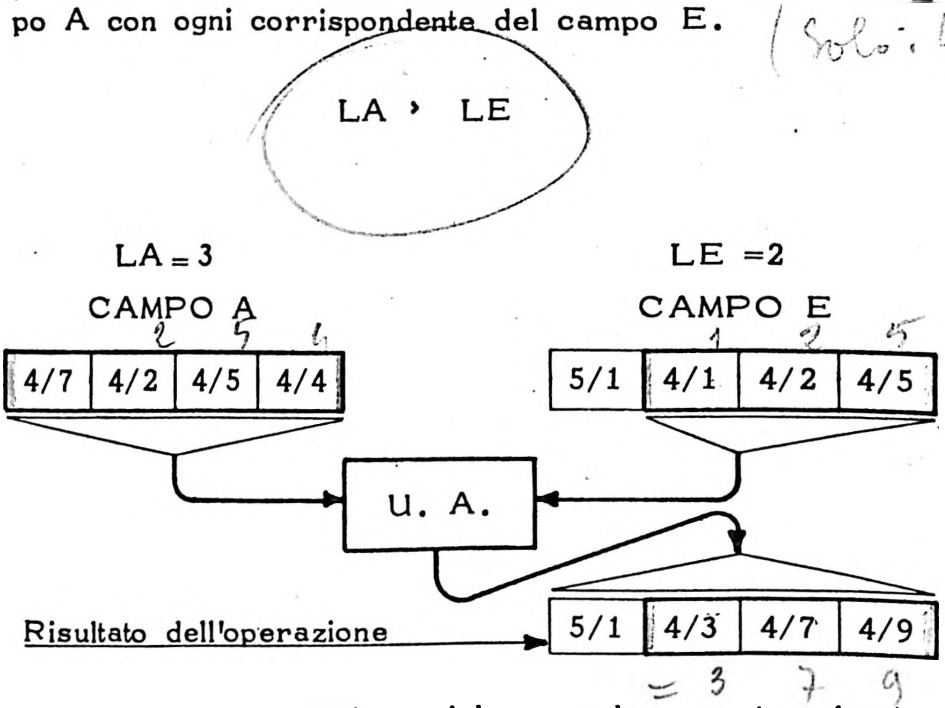
Il risultato va in campo E mentre il campo A resta immutato.

La parte più significativa di ciascun carattere del campo E rimane inalterata.

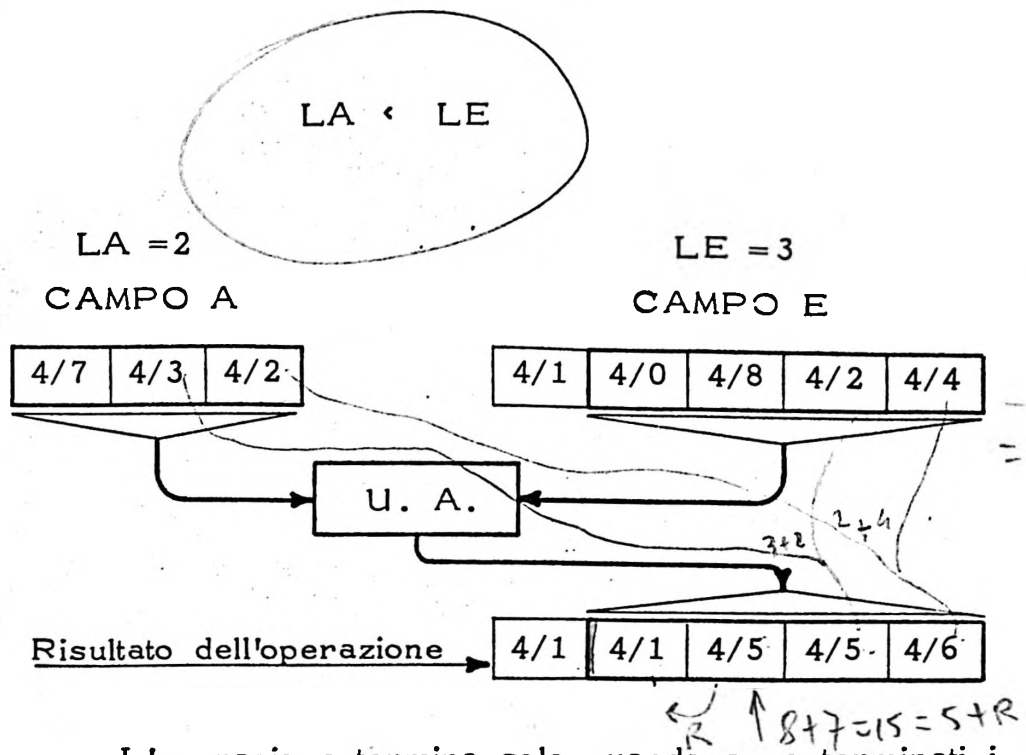
Riguardo a queste operazioni é necessario fare alcune osservazioni relative alla lunghezza dei campi.
Infatti si possono verificare tre casi :



In questo caso la fine elaborazione viene data dopo aver eseguito la somma di tutti i caratteri del campo A con ogni corrispondente del campo E.



La fine elaborazione si ha quando sono terminati i caratteri del campo E.



L'operazione termina solo quando sono terminati i caratteri del campo E (lunghezza LE), il campo A non viene più interessato; quando si é raggiunto la lunghezza LA si somma ad ogni successivo carattere del campo E uno zero.

L'ELABORATORE NON CONSIDERA NESSUN SEGNO ALGEBRICO.

8.3. In questo tipo di istruzioni é necessario tener presente :

- l'ordine di elaborazione degli operandi (indirizzi decrescenti)
- il numero dei caratteri da elaborare del campo A (LA)
- il numero dei caratteri da elaborare del campo E (LE)
- indirizzi iniziali dei campi
- fine dell'istruzione (LA > LE, LA = LE, LA < LE)
- eventuale risultato in campo E.

DOMANDE

Si consideri la seguente istruzione :

F/B	3/1	1/1	0/2	0/2	1/2

Si tenga presente che il carattere di funzione corrisponde ad una operazione di sottrazione decimale (S D).

In tale operazione si esegue E-A risultato in E.

a) - Quanti caratteri del campo A sono interessati ?

b) - Quanti caratteri del campo E sono interessati ?

c) - Qual'é l'indirizzo iniziale espresso in decimale del campo E ?

d) - Qual'é l'indirizzo iniziale, espresso in decimale, del campo A ?

e) - A quale indirizzo si trova l'ultimo carattere elaborato del campo E ?

f) - Quali bit, di ogni singolo carattere elaborato, vengono interessati ?

RISPOSTE

a) - Sono interessati 2 caratteri del campo A (LA = 1).

b) - 4 sono i caratteri interessati nel campo E poiché LE = 3.

c) - 4354.

d) - 530.

e) - 4351.

f) - Vengono interessati i bit meno significativi di ogni carattere elaborato. Tali bit sono detti anche bit numerici.

LEZIONE 9

Tempo di studio
40 minuti

9. ISTRUZIONI DEL GRUPPO A 1

9.1. ADDIZIONA IN DECIMALE

AD
F/A

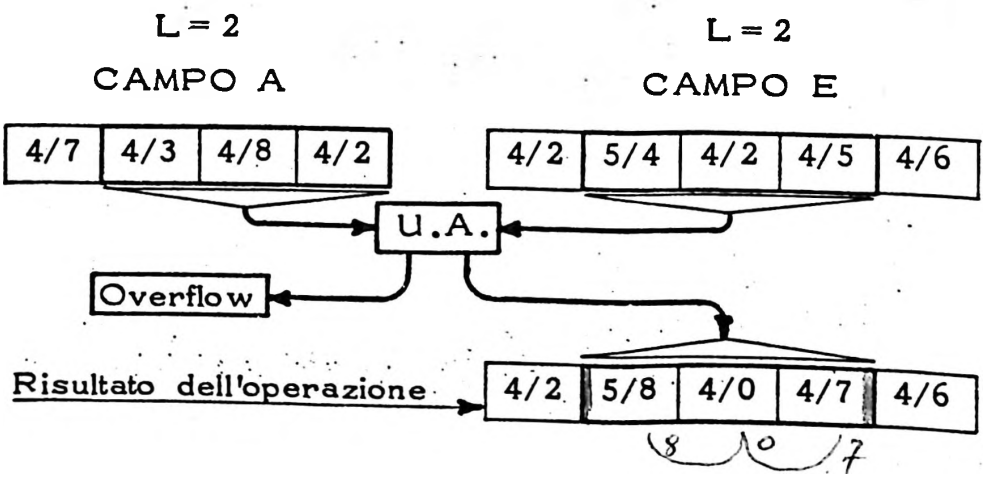
(Add Decimal).

Addizione decimale, tra il contenuto del campo E ed il contenuto del campo A. Opera solo sui 4 bit meno significativi di ogni carattere. Il risultato va in campo E, mentre il campo A resta immutato. Se LA > LE l'operazione interessa solo la parte meno significativa di A, per una lunghezza LE. Se LA < LE l'operazione continua per una lunghezza LE mentre A viene prolungato sommando zeri.
Non viene considerato alcun segno algebrico.

Se LA = LE fine elaborazione su fine lunghezza dei due campi (eventuale segnalazione di overflow (traboccamento) se l'operazione dà luogo ad un riporto che dovrebbe sommarsi con la posizione successiva alla fine del campo E.

Se LA < LE fine su lunghezza LE (aggiunta di zeri al campo A)

Se LA > LE fine su lunghezza LE (il risultato é incompleto).



9.2. SOTTRAE IN DECIMALE

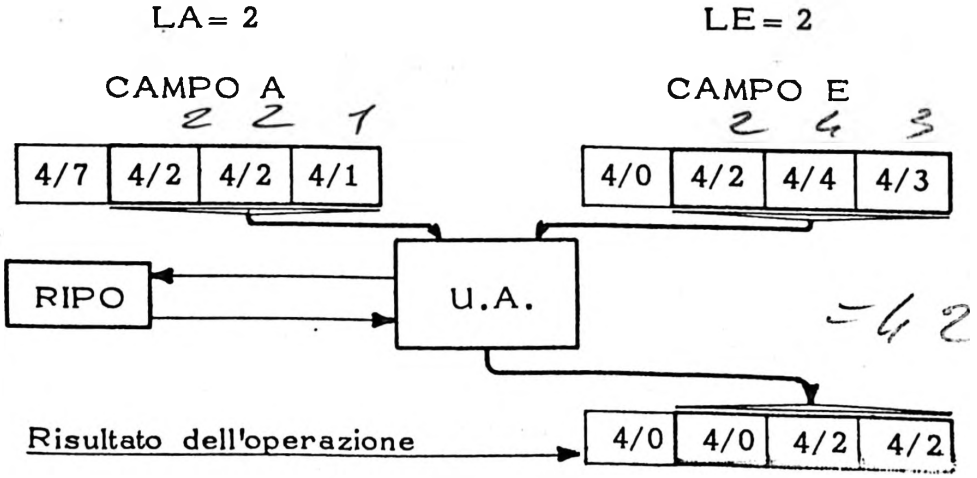
SD
F/B

(Subtract Decimal).

Sottrazione decimale, eseguita con le stesse modalità della AD. Gli operandi sono considerati entrambi positivi.
Si esegue :

E - A in E

Nel caso in cui il contenuto del campo E é minore di quello del campo A, il risultato in campo E si trova in complemento e NON viene raddrizzato automaticamente.
Cioé non esiste nell'elaboratore nessun dispositivo particolare che faccia ottenere il risultato con segno negativo.
Per questo motivo onde ottenere sempre un risultato reale si fa precedere l'istruzione SD da una istruzione di confronto onde stabilire quale dei due campi é il maggiore, cioé quale dovrà essere il campo E.



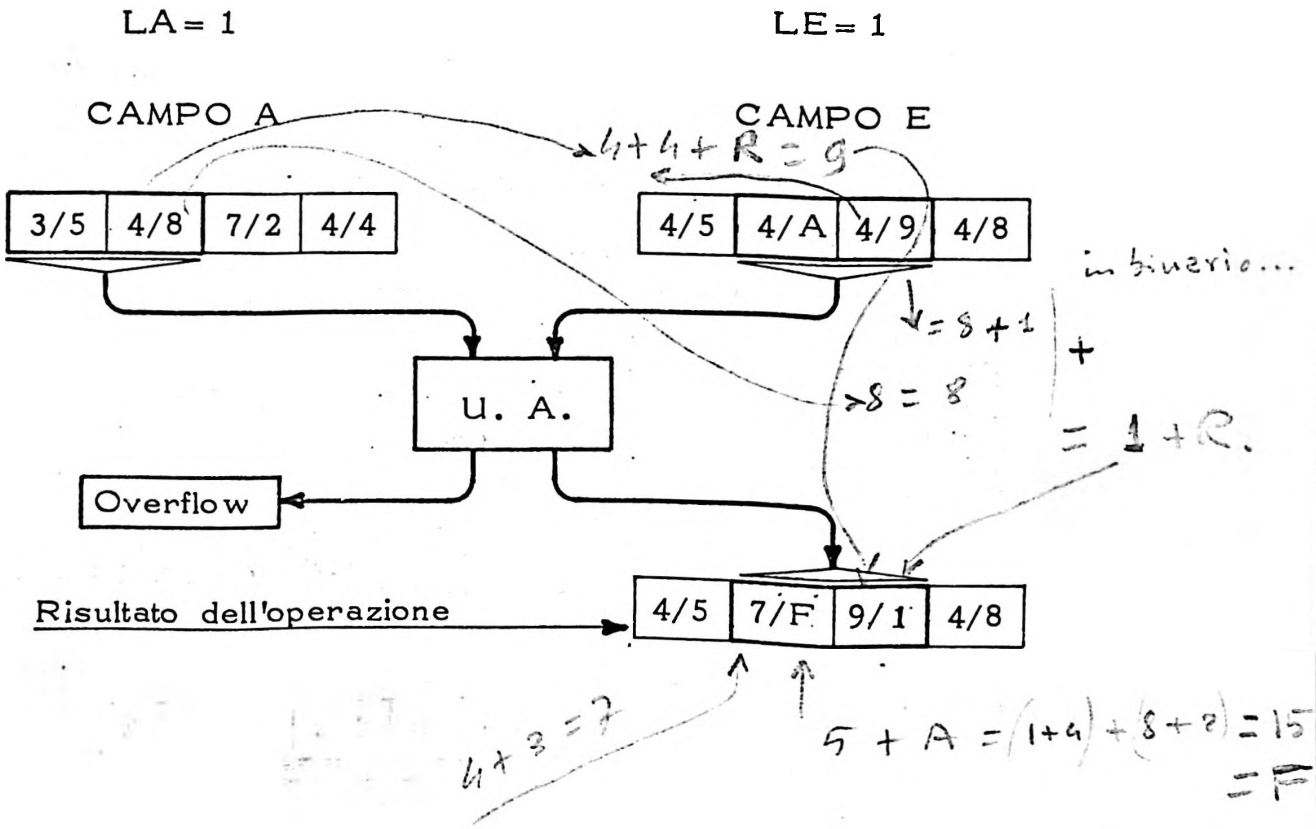
9.3. ADDIZIONA IN BINARIO

8 bit

AB
F/E

(Add Binary).

Addizione binaria eseguita con le stesse modalità della AD.
Vengono però sommati tutti gli 8 bit di ogni carattere.



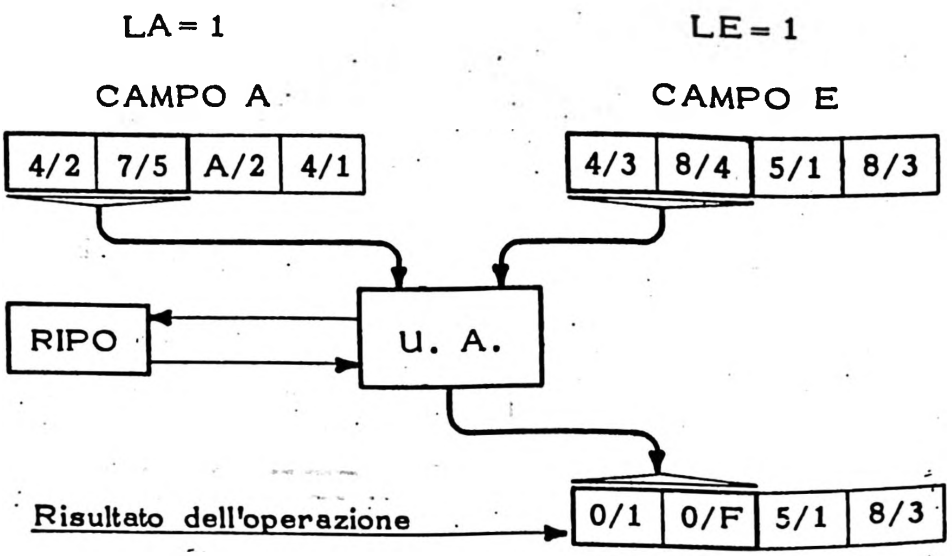
9.4. SOTTRAE IN BINARIO

SB
F/F

(Subtract Binary).

Sottrazione binaria eseguita con le stesse modalità della SD.

Vengono però sottratti tutti gli 8 bit di ogni carattere.



9.5. TRASFERISCE IN NUMERICO

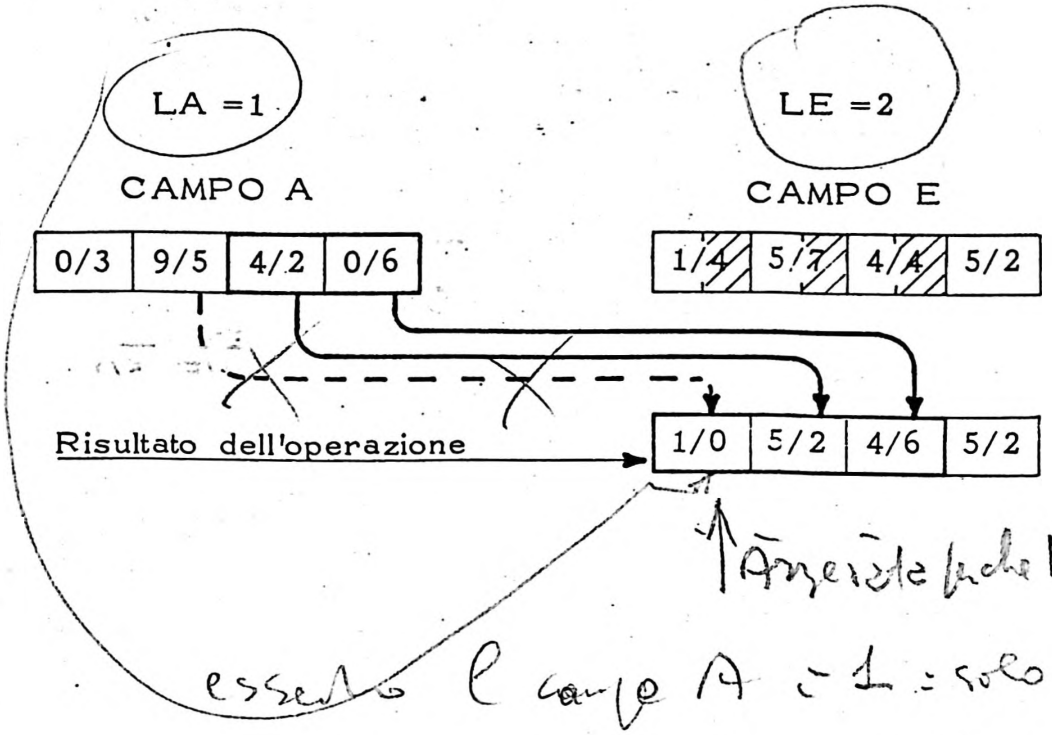
MVQ

(Move Right Quartets).

Trasferimento numerico del contenuto del campo A in campo E.

Vengono trasferiti solo i 4 bit meno significativi di ogni carattere, mentre non si considerano i 4 bit più significativi. Il campo A resta intatto.

L'operazione termina su lunghezza LE.
Se LE > LA il trasferimento viene completato con zeri.



Aggiorna finché LA = 1

essendo il campo A = 1 solo una posizione il resto va a zero la lunghezza campo E

9.6. CONFRONTA IN NUMERICO

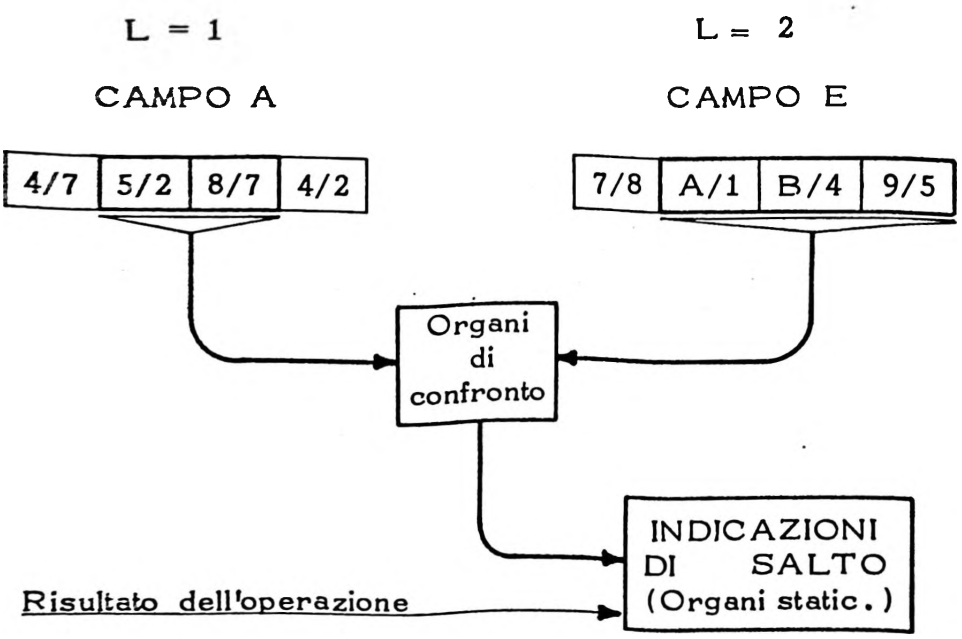
CMQ
F/9

(Compare Right Quartets).

Confronto numerico eseguito con le stesse modalità della AD.

Restano però intatti sia il campo A che il campo E. A fine operazione sono disponibili alcune indicazioni che servono come condizioni di salto programma. Esse sono :

E ‹ A; E = A; E › A



DOMANDE

a) - Che differenza esiste fra la MVC e la MVQ ? Nel campo A c. campo E

la M.V.C. trasferisce gli atleti = 8 bit. = numerico

la MVQ. sol. i 4 bit meno sign. cod. = Numerico

b) - Che differenza esiste fra la AB e la AD ?

la AB è addizione binaria su 8 bit.

la AD è addizione decimale numerica

c) - Che differenza esiste fra la CMC e la CMQ ? Campo E con campo A (4 bit) -

la CMQ compara i pari atleti meno sign. (4 bit)

la CMC gli atleti (tutti i bit)

RISPOSTE

a) - La MVC trasferisce tutti gli 8 bit di ogni carattere del campo A in campo E, la MVQ trasferisce solo i primi quattro bit di ogni carattere.
I primi 4 bit di un carattere sono i bit numerici.

b) - La AB somma tutti gli 8 bit di ogni carattere del campo E con i relativi 8 bit del campo A, la AD somma solo i primi 4 bit di ogni carattere dei due campi.

c) - La CMC confronta gli 8 bit di ogni carattere del campo E con i relativi 8 bit del campo A, la CMQ compara solo i primi 4 bit di ogni carattere dei due campi.

LEZIONE 10

Tempo di studio
20 minuti

10. ISTRUZIONI DEL GRUPPO B3

10.1. CARATTERISTICHE COMUNI ALLE ISTRUZIONI DEL GRUPPO B3

A questo gruppo appartengono due istruzioni che sono composte da soli quattro caratteri.

Esse si presentano con i seguenti codici :

F A IE^+ IE^-

In queste istruzioni il carattere seguente quello di funzione, non rappresenta la lunghezza dell'operando, bensì il carattere che l'istruzione stessa deve trasferire o confrontare con quello posto all'indirizzo espresso con IE^+ IE^-.

Le caratteristiche principali sono :

FASE DI ESECUZIONE

Si elabora un solo carattere, quello situato nell'istruzione stessa al posto della lunghezza.

INDIRIZZO INIZIALE DEI CAMPI

Si esprime un solo indirizzo quello del campo E.

LUNGHEZZA DEGLI OPERANDI

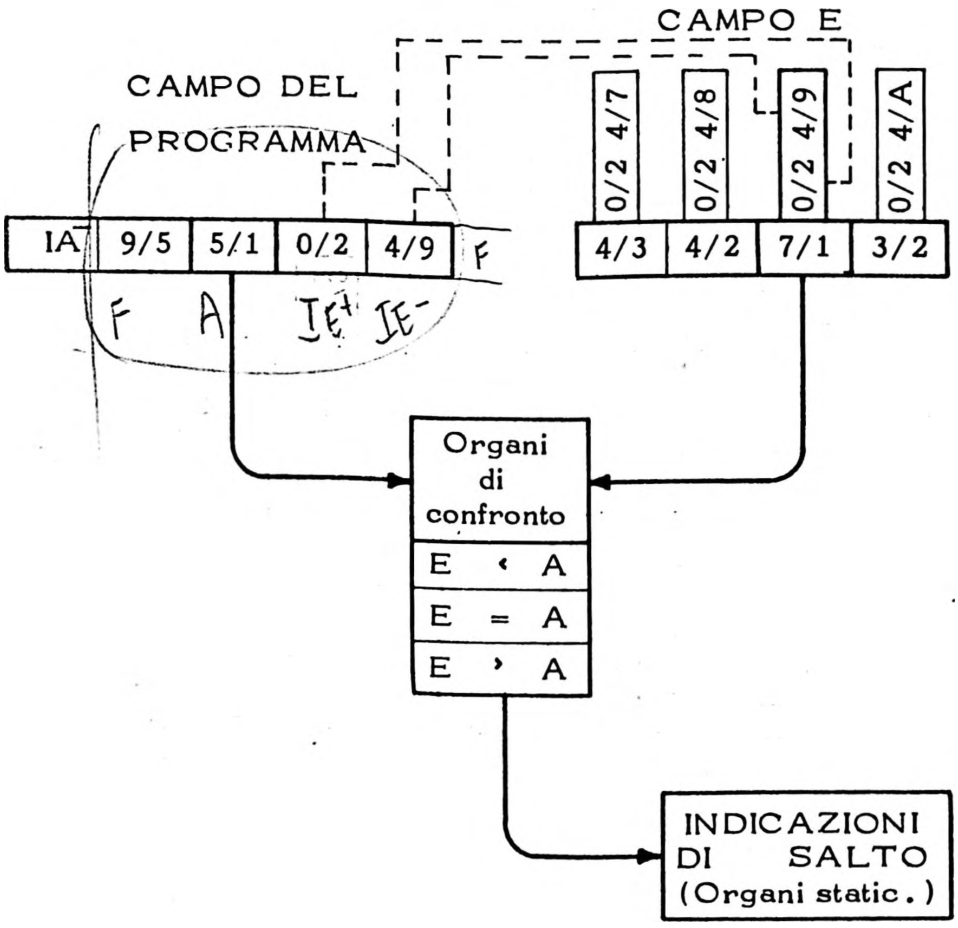
Non esiste. Il carattere che si trova in questa posizione non rappresenta la lunghezza dei dati da elaborare bensì l'operando o uno dei due operandi dell'operazione.

10.2. CONFRONTA MEMORIA CON L'IMMEDIATO

CMI
9/5

(Compare Immediate to Store).

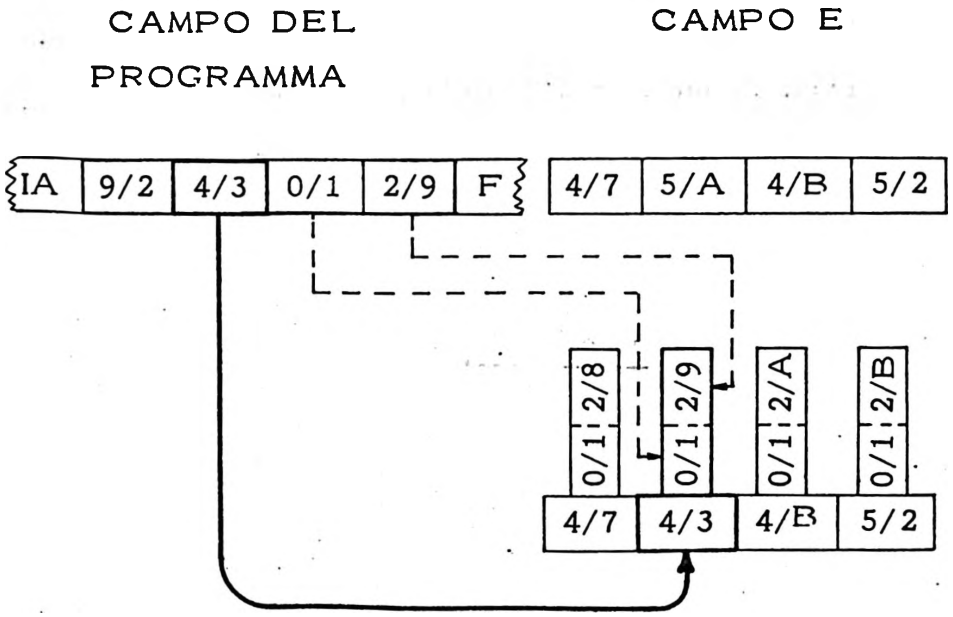
Confronto tra il carattere A ed il carattere che si trova scritto all'indirizzo IE.
Sono disponibili le seguenti condizioni di salto :



10.3. TRASFERISCE L'IMMEDIATO

MVI 9/2 (Move Immediate Octet).

Trasferimento del carattere A all'indirizzo IE.
Le condizioni di salto restano immutate.



DOMANDE

a) - Che differenza esiste rispetto al numero di caratteri che le compongono, tra le istruzioni del gruppo A1 e A2 e quello del gruppo B3 ?

Per A1 e A2 ..messaggio.. nel gruppo B3 = 4 caratteri

b) - Qual'é il significato che assume la lunghezza nei vari gruppi di istruzione ? Nei gruppi A1-A2 la lunghezza che segue quella di funzione determina la lunghezza dei campi - nel B3 rappresenta il carattere da elaborare

c) - Quanti caratteri si elaborano nelle istruzioni di tipo B3 ? 1 carattere

RISPOSTE

a) - Le istruzioni appartenenti al gruppo A1 e A2 sono formate da 6 caratteri. Quelle appartenenti al gruppo B3 sono formate da 4 caratteri.

b) - Nei gruppi A1 e A2 le istruzioni hanno il carattere che segue quello di funzione che rappresenta la lunghezza degli operandi. Nel gruppo B3 tale carattere rappresenta un carattere da elaborare.

c) - Si elabora un solo carattere.

LEZIONE 11

Tempo di studio
30 minuti

11. ISTRUZIONI DEL GRUPPO C4

11.1. CARATTERISTICHE COMUNI ALLE ISTRUZIONI DEL GRUPPO C4

A questo gruppo appartengono istruzioni che svolgono un'operazione chiamata SALTO (JUMP).

Tale istruzione é formata da :

F carattere di funzione

F' carattere particolare variabile da tipo di salto a tipo di salto

IA indirizzo dal quale si riprende il programma se la condizione F' si é verificata in una precedente elaborazione.

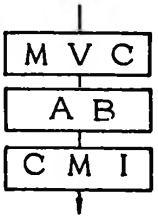
Il modo di utilizzare in un programma le istruzioni di salto viene analizzato dettagliatamente nelle pagine seguenti.

11.2. UTILIZZAZIONE DEI SALTI

Nel caso di una elaborazione si possono verificare o non verificare determinati eventi come : l'eguaglianza o la disuguaglianza dopo un confronto tra i caratteri appartenenti ai due campi, un'operazione aritmetica che dà come risultato zero, ecc.

Si può controllare se si sono verificati questi "avvenimenti" mediante le istruzioni di salto.

Si consideri ora il seguente programma composto da tre istruzioni :



Si supponga che queste istruzioni eseguano le operazioni sotto indicate :

- la MVC, un'operazione di trasferimento dati per creare il campo E della AB subito seguente

- la AB, la somma binaria tra l'operando del campo E con quello del campo A

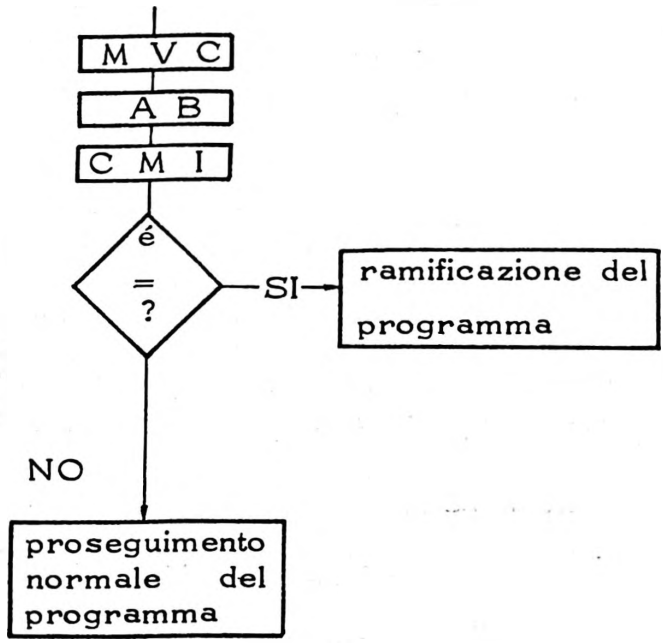
- la CMI, il confronto del carattere A, contenuto nell'istruzione stessa, con l'ultimo carattere elaborato nel campo E della operazione AB.

A conclusione di questa semplice elaborazione si possono presentare due casi distinti: che l'istruzione di confronto abbia dato segnalazione di risultato.

UGUALE o DIVERSO

Ora, bisogna prevedere cosa l'elaboratore deve fare se si verifica un caso o l'altro. E' necessario cioé che dopo l'istruzione CMI ci sia un'altra istruzione capace di controllare e utilizzare l'evento che si é verificato.

Se si vuole utilizzare il risultato del confronto il programma sopra analizzato deve essere integrato nel seguente modo :



Cioé si deve introdurre dopo la CMI un'istruzione di SALTO.

Infatti, terminata la CMI l'elaboratore dispone nel suo interno del risultato del confronto ed esegue l'istruzione di salto che può dar luogo o no al salto.

Se quest'ultimo si verifica (risultato del confronto UGUALE) si eseguono le istruzioni che compongono una ramificazione del programma. Se invece il salto non si verifica (risultato del confronto DIVERSO)il programma prosegue la sua sequenza normale.

11.3. CARATTERISTICHE PRINCIPALI DEI SALTI :

FASE DI ESECUZIONE

I salti non hanno una vera e propria fase di elaborazione dati (istruzione senza fase esecutiva).

LUNGHEZZA DEGLI OPERANDI

Non esiste. Il carattere che si trova in questa posizione non rappresenta la lunghezza dei dati da elaborare ma un carattere (F') che indica quale condizione deve essere presente perché il salto si verifichi.

INDIRIZZO INIZIALE DEI CAMPI

L'indirizzo espresso nell'istruzione non si riferisce ad un campo di elaborazione bensì alla posizione di programma alla quale si vuole saltare se l'elaboratore dà la segnalazione di condizione verificata.

RISULTATO

Il risultato può essere diverso a seconda che il salto sia :

1) - non verificato

2) - verificato

Nel primo caso non si ha nessuna modifica allo svolgimento normale del programma.

Nel secondo caso il programma salta all'indirizzo espresso nell'istruzione stessa.

E' necessario ricordare che :

DOPO UN SALTO VERIFICATO LA POSIZIONE DI MEMORIA DALLA QUALE SI CONTINUA L'ELABORAZIONE, DEVE CONTENERE SEMPRE UN CODICE DI FUNZIONE DI UNA ISTRUZIONE.

Ciò significa che l'indirizzo che compare nell'istruzione di salto deve sempre inviare ad un codice di funzione dell'istruzione dalla quale si vuole proseguire il programma.

DOMANDE

a) - L'istruzione di salto che operazione svolge ?
trasferisce il salto da una posizione a cui altrimenti non proseguirebbe

b) - Il salto quando viene eseguito ?
se quella determinata è l'istruzione di salto e questa si verifica la condizione

c) - Il salto come può essere predisposto ?
A/a una congruenza / la condizione = ≠ ecc

d) - Qual'é il codice contenuto nella posizione di memoria dalla quale si continua l'elaborazione dopo l'istruzione di salto ? un codice di funzione

RISPOSTE

a) - Svolge un'operazione che consente di saltare da una posizione ad un'altra del programma.

b) - Il salto viene eseguito quando il programma incontra la istruzione di salto e quando nell'elaboratore si é verificata la condizione per la quale il salto é stato predisposto.

c) - Per saltare su condizione di eguaglianza o di diseguaglianza oltre che per altre condizioni che si vedranno in seguito.

d) - Deve essere sempre un codice di funzione.

LEZIONE 12

Tempo di studio
1 ora

12. ISTRUZIONI DI SALTO NORMALE

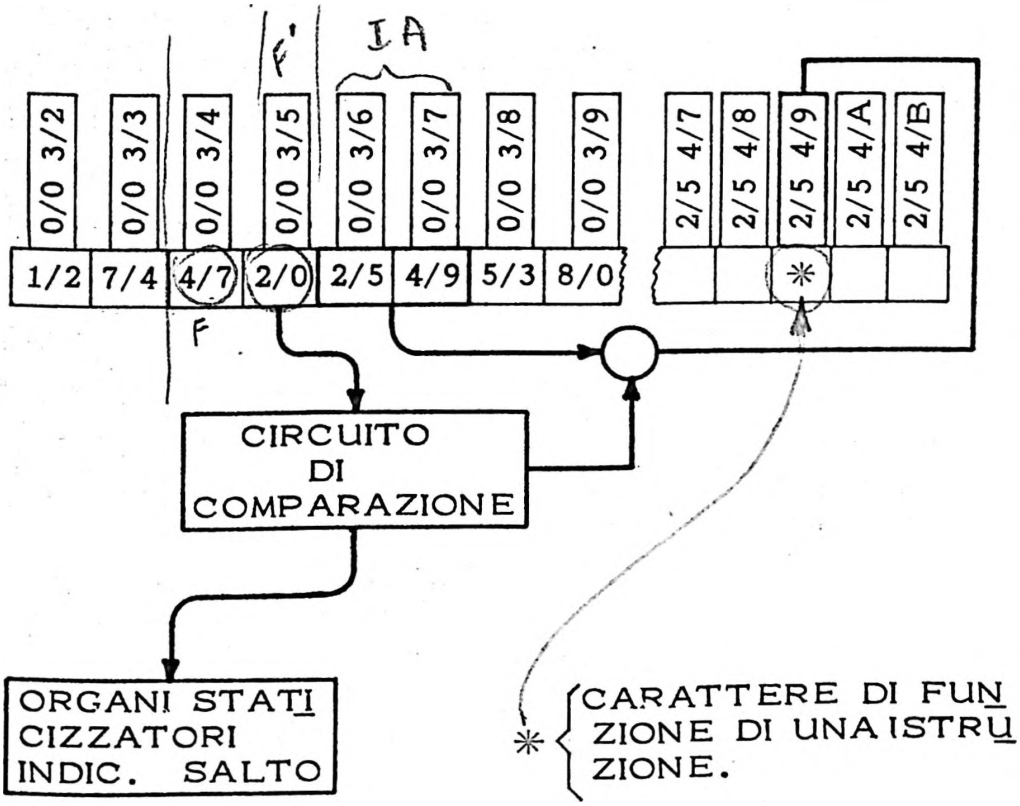
12.1. SALTO CONDIZIONATO

J C
4/7

(Jump condition).

L'istruzione di salto condizionato verifica se in precedenza é avvenuto un determinato evento :

- se questo evento si é verificato essa prosegue il programma saltando all'indirizzo espresso nella stessa istruzione
- se questo evento NON si é verificato prosegue normalmente il programma analizzando l'istruzione subito successiva.



12.2. TIPI DI SALTO CONDIZIONATO

La prima istruzione di salto che si può considerare é la JC così composta :

F F' IA^+ IA^-

Il codice F' di questa istruzione può assumere diversi valori che corrispondono alle diverse condizioni che si possono verificare durante l'evoluzione di un programma.

I codici F' sono :

	8/0	4/0	2/0	1/0
AD, AB	E + A = 0	E + A ≠ 0	Overflow E + A = 0	Overflow E + A ≠ 0
SD, SB		E - A < 0	E - A = 0	E - A > 0
Confronti		E < A	E = A	E > A
MVQ	E = 0	E ≠ 0		
OC			E ⊕ A = 0	E ⊕ A ≠ 0
SL - SR			Crt. non trovato	Crt. trovato

Si può osservare che i bit di F' che interessano tale codice sono i bit più significativi.

Se l'istruzione di salto ha come F' 8/0 si ottiene che se essa segue una AD o AB, verificherà se il risultato di questa operazione é uguale a zero.

Se segue una MVQ verificherà se quest'ultima trasferisce in campo E una parte numerica uguale a zero.

In relazione al valore che si attribuisce ad F' i salti condizionati assumono i seguenti nomi :

Denominazione	Descrizione	F	F'
JE	Salta confronto uguale (Jump if Equal)	4/7	2/0
JG	Salta se confronto maggiore (Jump if Greater)	4/7	1/0
JGE	Salta se confronto maggiore o uguale (Jump if Greater or Equal)	4/7	3/0
JL	Salta se confronto minore (Jump if Less)	4/7	4/0
JLE	Salta se confronto minore o uguale (Jump if Less or Equal)	4/7	6/0
JNE	Salta se confronto diverso (Jump if Not Equal)	4/7	5/0
JU	Salta incondizionatamente (Jump Unconditional)	4/7	F/0

fig. 12.2.1

DOMANDE

Tenendo presente la tabella dei valori di F' di figura 12.2.1 rispondere alle seguenti domande :

a) - Dopo una istruzione di ricerca SR si vuole verificare se la istruzione é terminata perché si é trovato il carattere voluto. Se ciò avviene, si vuole arrestare l'elaborazione senza eseguire tutte le istruzioni seguenti la SR. Qual'é il valore del carattere F' da inserire nella istruzione di salto ?

1/0

b) - Si vuole verificare se una SB ha dato come risultato E - A > 0 indi ritornare in "testa" al programma. Qual'é il codice F' che si deve mettere nella istruzione di salto che segue la SB ?

1/0

c) - Si abbia un'istruzione di confronto seguita da un JC così compilata :

4/7 5/0 IA$^+$ IA$^-$

per quale condizione si verificherà il salto ?

Salta se confronto ≠ e continua il comando di Fondione all'IA+IA=

d) - Dopo una istruzione di sottrazione, si vuol controllare se il risultato NON sia E - A < 0. Se ciò si verifica si vuol saltare un certo gruppo di istruzioni. Che codice F' bisognerà mettere nell'istruzione JC che segue la sottrazione ?

3/0

e) - Se in un' istruzione di salto il codice F' é uguale a F/0, per quale condizione si verificherà il salto ?

Salta sempre senza condizione alcuna

f) - Se in una istruzione di salto il codice F' é uguale a 0/0, per quale condizione si verificherà il salto?

Mai salta perché una condizione

= > < ≠ c'è sempre e

Ho condizione 0/0 mai.

RISPOSTE

a) - La ricerca dovrà essere seguita da un istruzione di salto il cui F' = 1/0.

b) - L'istruzione di salto dovrà avere F' = 1/0.

c) - IL SALTO SI VERIFICA PER E ≠ A. INFATTI E' COME SE IL SALTO AVESSE DUE F' UNO UGUALE A 4/0 (E‹ A) E L'ALTRO UGUALE 1/0 (E ›A). QUINDI L'OR DEI BIT DEL CODICE F' PUO' CREARE NUOVE POSSIBILITA' DI CODICI DI F'.

d) - Il codice F', dell'istruzione di salto seguente la sottrazione, che segnali che NON si ha E - A ‹ 0 é F' 3/0 che corrisponde a (E - A = 0)+(E - A ›0).

e) - IL SALTO SI VERIFICHERA' PER TUTTE LE CONDIZIONI. In pratica il salto é incondizionato poi ché almeno una delle condizioni esiste sempre. Quindi una JC così compilata :

4/7 F/0 IA^+ IA^-

esegue sempre il salto, cioé si ha un salto incondizionato. Tale salto assume il nome JU (Jump Un conditional).

f) - Affinché un salto così codificato si verifichi occorre che NESSUNA delle condizioni riportate nella tabella di fig. 12.2.1 si manifesti. Siccome ciò é impossibile il salto non si verificherà mai. Per esempio il risultato di una comparazione sarà MAGGIORE, o MINORE o EGUALE ma in nessun caso si avrà la assenza contemporanea delle tre segnalazioni.

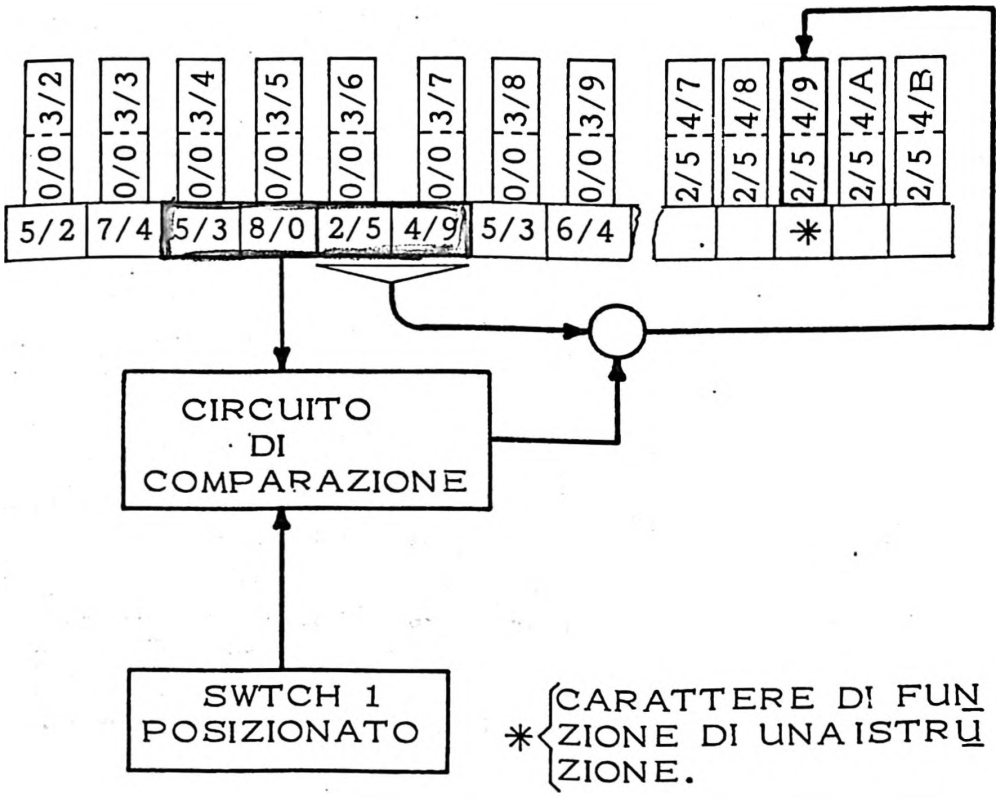
12.3. SALTO CONDIZIONATO AL POSIZIONAMENTO DEI PULSANTI 1 O 2 DI CONSOLE

J S
5/3

(Jump on Switch 1 or 2).

Questa istruzione di salto ha lo scopo di controllare in quale posizione si trova il pulsante 1 o il pulsante 2 posti sulla console operativa.

Tali controlli si effettuano tramite il valore che si attribuisce al carattere F'.



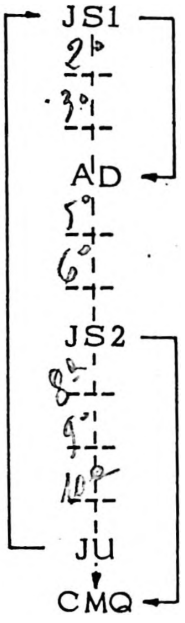
12.4. I valori che in queste istruzioni si devono attribuire ad F' sono :

F'			
8/0	4/0	2/0	1/0
SWITCH 1 (COES 1)	SWITCH 2 (COES 2)	ERIR	COEN

Dove 8/0 indica che il salto si verifica se il pulsante bistabile SWITCH 1 posto sulla console, é posizionato a lavoro. Così pure il valore F' 4/0 che però controlla il posizionamento a lavoro del pulsante bistabile SWITCH 2 di console.

Questi due pulsanti consentono, quando i relativi salti sono contenuti in un programma, di far si che il programma stesso modifichi la sua sequenza.

Si abbia il seguente programma esemplificativo :



Si possono verificare tre casi :

1) - Se i pulsanti SWITCH 1 e SWITCH 2 sono a riposo il programma si svolge secondo la linea tratteggiata.

2) - Se lo SWITCH 1 é posto a lavoro il programma non esegue la 2^ e 3^ istruzione e inizierà il lavoro della AD.

3) - Se proseguendo lo svolgimento del programma l'elaboratore si trova con il pulsante posto in console, SWITCH 2 abbassato si salterà direttamente all'istruzione CMQ se invece il pulsante SWITCH 2 non é a lavoro si eseguiranno anche le istruzioni subito seguenti alla JS 2 per poi riprendere il programma dall'inizio e ripeterlo.

Come si può osservare agendo manualmente sui due pulsanti posti in console si ha la possibilità di modificare lo svolgimento di un programma.

Questo tipo di istruzione é utilizzato anche per controllare se si sono verificate le condizioni ERIR e COEN.

- La condizione ERIR, esplorabile con F'= 2/0, é la segnalazione che dà l'elaboratore quando si é verificata una anormalità di funzionamento nei circuiti della memoria.

- La condizione COEN, esplorabile con F'= 1/0, é la segnalazione che in precedenza é stata eseguita una istruzione LON il cui significato sarà visto in seguito.

DOMANDE

a) - Si può variare lo svolgimento di un programma agendo manualmente da console ?
Con i pulsanti 1 e 2 applicando una opportuna di salto JS1 - o JS2

b) - Quando compare la segnalazione d'errore ERIR come si può agire da programma ? Con JS F'=2/0
Case si fa???

c) - Il segnale COEN può essere utilizzato con un salto ?
JS con F'=1/0 ...?

RISPOSTE

a) - Si purché nel programma siano inserite le istruzioni JS 1 (SWITCH 1) o JS 2 (SWITCH 2) corrispondenti ai pulsanti posti in console.

b) - Si può agire avendo previsto ciò e avendo posto una JS con F' = 2/0.

c) - Si può essere utilizzato con una istruzione JS avente F' = 1/0.

LEZIONE 13

Tempo di studio
20 minuti

13. ISTRUZIONE DI SALTO PARTICOLARE

13.1. SALTO CON MEMORIZZAZIONE DELL'INDIRIZZO DI INTERRUZIONE

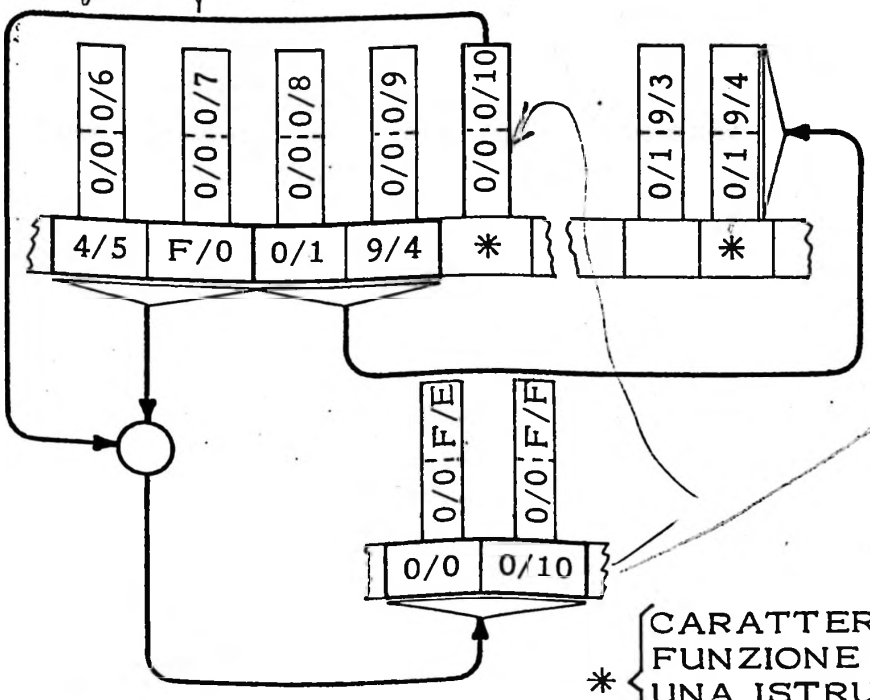
JRT
4/5

(Jump And Return).

Tale istruzione ha il codice di funzione F' = F/0 e quindi si comporta come un salto incondizionato.

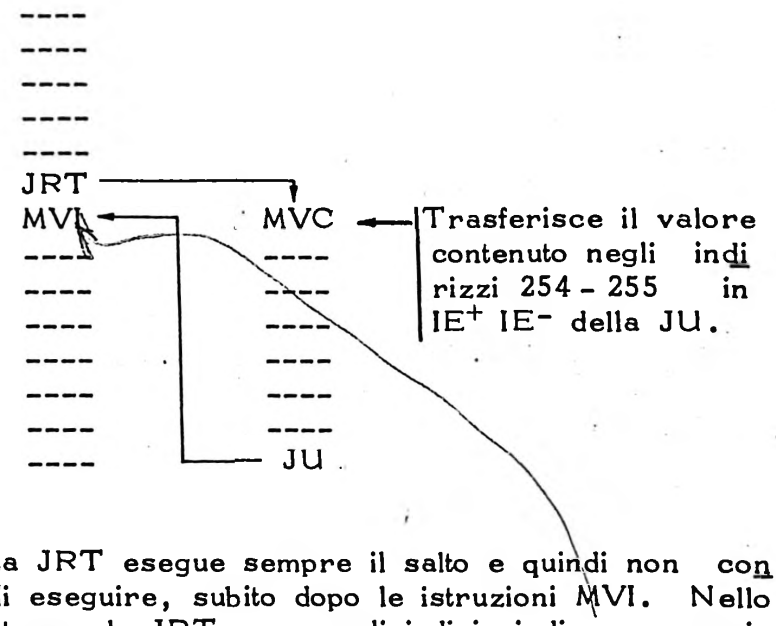
Presenta la caratteristica, oltre a quella di saltare incondizionatamente, di memorizzare un indirizzo.

Più precisamente immagazzina negli indirizzi di memoria 255 e 254 l'indirizzo del codice di funzione dell'istruzione che segue la JRT.



* CARATTERE DI FUNZIONE DI UNA ISTRUZIONE

Per esempio nel seguente programma :



La JRT esegue sempre il salto e quindi non consente di eseguire, subito dopo le istruzioni MVI. Nello stesso tempo la JRT pone negli indirizzi di memoria 254 - 255 l'indirizzo del codice di funzione dell'istruzione che la segue cioé della MVI.

Questa operazione é molto utile qualora si voglia riprendere il programma dal punto in cui si era interrotto dopo aver svolto altre operazioni. In definitiva la JRT può sostituirsi alla JU, cioé al salto incondizionato, tutte le volte che interessa saltare senza condizioni particolari ma ricordare il punto dal quale si vuole riprendere la sequenza interrotta del programma.

In pratica nella JRT si verifica che :

- l'indirizzo memorizzato può essere utilizzato per compilare l'indirizzo di qualche altra istruzione (ad esempio un salto) in modo da poter riprendere il programma dal punto in cui si era interrotto.

DOMANDE

a) - Che operazione diversa dagli altri salti svolge la JRT ? salta in continuazione come JU
.....(ma memorizza)
.....

b) - Cosa si trova negli indirizzi 254 - 255 di memoria dopo che si é eseguita una JRT ?
L'indirizzo del Byte di Funzione della linea istruzione
che segue la JRT.
.....

c) - E' utilizzabile il contenuto delle posizioni 254 - 255?
Sé è la completa
l'indirizzo di una altra istruzione es. JU
.....

d) - Quando é conveniente sostituire una JU con una JRT ? Per ritornare al programma nel
punto dove si era lasciato -
.....

RISPOSTE

a) - Oltre che ad effettuare un salto incondizionato la JRT pone l'indirizzo del codice di funzione del l'istruzione che la segue negli indirizzi 254 - 255 di memoria.

b) - L'indirizzo del codice di funzione della istruzione che segue la JRT.

c) - Si può utilizzare per compilare l'indirizzo di una altra istruzione.

d) - E' conveniente sostituire una JU (incondizionata) con una JRT quando oltre ad effettuare il salto si desidera ricordare la posizione di memoria in cui si trova il codice di funzione dell'istruzione che la segue.

LEZIONE 14

Tempo di studio
30 minuti

14. ISTRUZIONI DEL GRUPPO D4

14.1. CONSOLE OPERATIVA

Prima di esaminare le istruzioni appartenenti al gruppo D4, occorre conoscere un particolare organo dell'elaboratore:

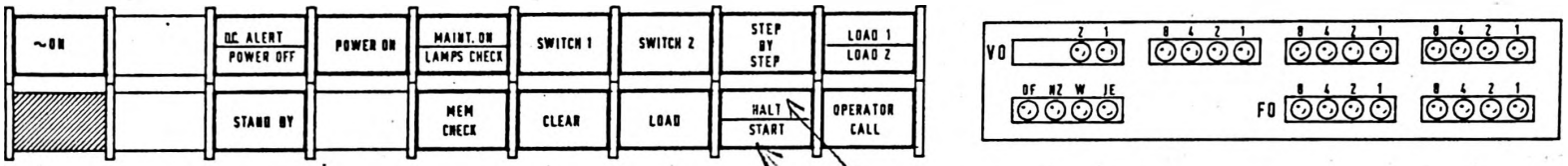
LA CONSOLE OPERATIVA

Questo organo, rappresentato in fig. 14.1.1 é costituito da pulsanti, interruttori e segnalazioni luminose.

Dalla console si possono compiere diverse operazioni :

- di carattere generale come la messa sotto tensione dell'elaboratore
- di avvio della elaborazione, cioé di introduzione del programma nell'elaboratore
- di condizionamento di alcune istruzioni del programma
- di arresto del programma

La funzione specifica dei pulsanti, degli interruttori e delle segnalazioni luminose é spiegata nel testo "U.C. Vol. 1°". Per ora si esamina soltanto l'utilizzazione degli organi di console che interessano le istruzioni.

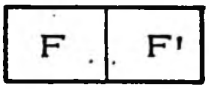


CONSOLE OPERATIVA GE 115

fig. 14.1.1

14.2. CARATTERISTICHE COMUNI ALLE ISTRUZIONI DEL GRUPPO D4

Il gruppo D4 é formato da istruzioni composte da due soli caratteri.



F carattere di funzione

F' carattere particolare che esamina una certa condizione

Le caratteristiche comuni sono :

FASE DI ESECUZIONE

Queste istruzioni non hanno nessuna fase di esecuzione.

INDIRIZZO INIZIALE DEI CAMPI

In queste istruzioni gli indirizzi non sono espressi perché non interessano.

LUNGHEZZA DELL'OPERANDO

Il carattere che segue il codice di funzione non esprime la lunghezza dei dati da elaborare ma rappresenta un carattere chiamato "COMPLEMENTO DEL CODICE DI FUNZIONE", che ha lo scopo di indicare quale operazione si deve eseguire.

RISULTATO

In genere il risultato non é rappresentato da un numero o da un carattere posto in memoria ma da una particolare operazione fisica (accendi lampadina, disabilita comandi ecc).

14.3. ISTRUZIONE DI ARRESTO DEL PROGRAMMA

HLT
0/A

(Halt System Operation).

E' l'istruzione che, una volta eseguita, arresta l'elaborazione. Essa é formata dai seguenti due caratteri :

F	F'
0/10	0/0

Se tale istruzione é seguita, come normalmente avviene, da altre istruzioni é sufficiente premere il pulsante START per far si che il programma riprenda da questo punto il suo svolgimento.

14.4. ISTRUZIONE DI NON OPERAZIONE

NOP 2
0/7

(No operation).

Tale istruzione non esegue alcuna operazione.

Essa ha il compito di occupare due posizioni di memoria che potrebbero venire utilizzate durante la messa a punto o la revisione di un programma.

Essa é composta dai seguenti due caratteri :

F	F'
0/7	0/0

14.5. ISTRUZIONE ACCENDE LUCE "OPERATOR CALL"

LON (Turn Alert Light on).

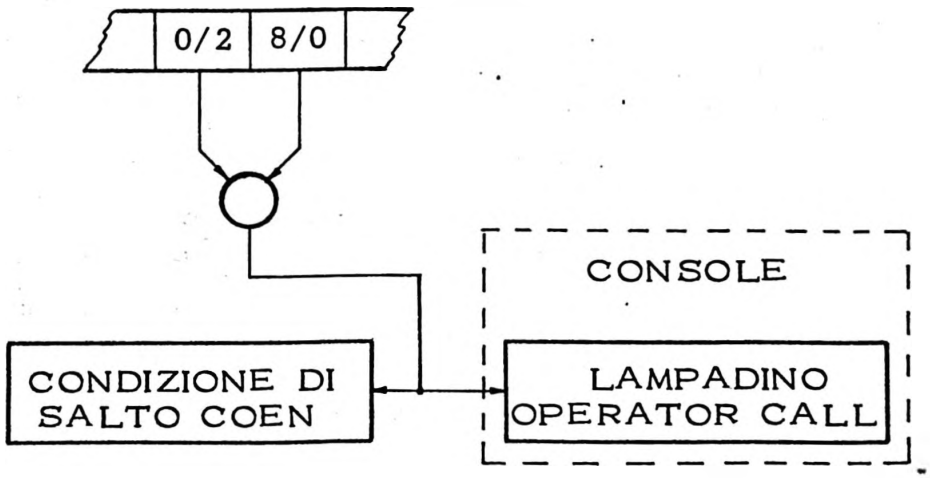
0/2 Codice di funzione F' 8/0.

Accende il lampadino chiamato "OPERATOR CALL", situato sulla console, e predispone la condizione di salto COEN. Tale condizione é esplorabile con un salto codificato

5/3 1/0 IA+ IA-

L'istruzione LON si dimostra utile quando necessita segnalare all'operatore che il programma si é svolto secondo una certa sequenza (quella contenente la LON) oppure un'altra (che non contiene la LON).

LON



14.6. ISTRUZIONE SPEGNE LUCE "OPERATOR CALL"

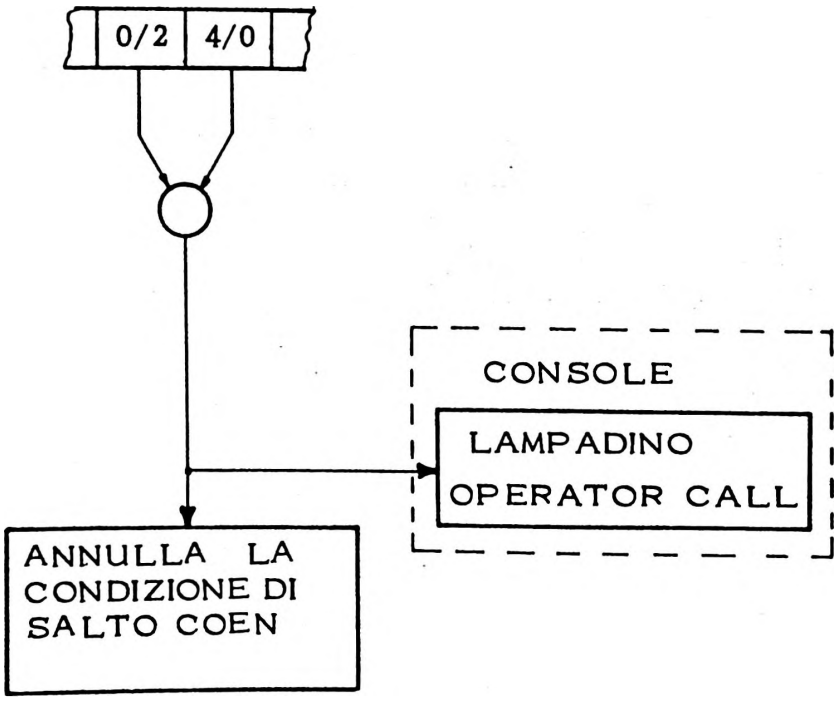
LOFF
0/2

(Turn Alert Light Off).

Codice di funzione F' = 4/0

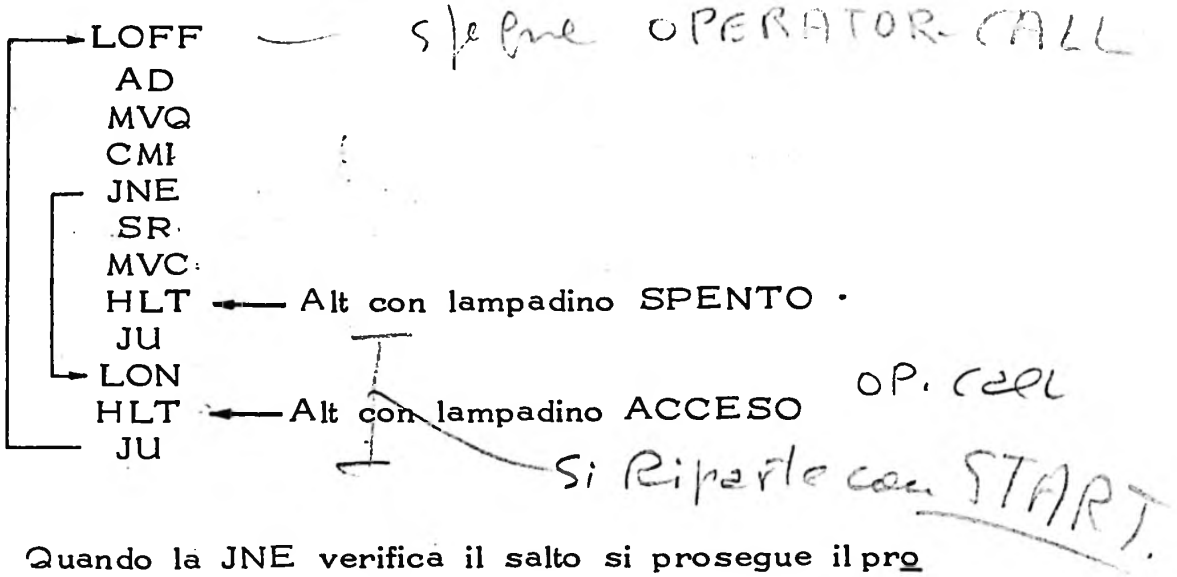
Spegne la lampada chiamata "OPERATOR CALL" situata sulla console (accesa da una istruzione LON) ed annulla la condizione di salto COEN.

LOFF



14.7. UTILIZZAZIONE DELLE ISTRUZIONI LON E LOFF

Si analizza ora la seguente sequenza di istruzioni :



Quando la JNE verifica il salto si prosegue il programma eseguendo la LON, che accende la lampadina sulla console.

Subito dopo si esegue l'istruzione HLT che fa arrestare l'elaborazione. Quindi la segnalazione luminosa indica, oltre l'arresto dell'elaboratore, che questa é avvenuta perché la JNE ha verificato la condizione di salto. Nel caso che l'elaboratore si arresti senza segnalazione luminosa significa che JNE non ha verificato la condizione di salto.

Nel caso descritto l'accensione della segnalazione luminosa indica che l'operazione di salto si é verificata.

Dopo ogni HLT é possibile riprendere l'elavorazione premendo il pulsante START posto sulla console.

In tal caso si prosegue l'elaborazione eseguendo la JU che inserisce la LOFF che spegne la segnalazione luminosa.

DOMANDE

a) - Se una istruzione HLT é seguita da altre istruzioni come si deve procedere per far proseguire il programma ?

con test seguito - START

b) - Quale scopo ha inserire delle istruzioni NOP 2 in un programma ?

NOP+NOP+NOP

si tenere due istruzioni 2+2+2+2 ecc

a fine se tu vuoi fare a mano a mano a fine preciso

c) - L'accensione del lampadino di console "OPERATOR CALL" cosa indica ?

che la memoria salta LOW

si è verificato

d) - Quale istruzione é necessario eseguire per spegnere la segnalazione luminosa " OPERATOR CALL" ?

l'istruzione LOFF reset. sulla

con JU (salto)

RISPOSTE

a) - E' sufficiente premere il pulsante START posto sulla console operativa.

b) - L'inserimento delle NOP 2 in un programma ha lo scopo di mantenere delle posizioni disponibili per eventuali modifiche che lo stesso programma può subire.

c) - Indica che il programma ha eseguito una istruzione LON.

d) - E' necessario eseguire l'istruzione LOFF.

LEZIONE 15

Tempo di studio
20 minuti

Passo a passo

15. ISTRUZIONI RELATIVE AL PULSANTE STEP BY STEP

15.1. PULSANTE "STEP BY STEP".

Sulla console operativa esiste un pulsante denominato STEP BY STEP che permette di arrestare lo svolgimento del programma dopo l'esecuzione di ogni singola istruzione. Si può eseguire così una sola istruzione ad ogni premuta sul pulsante "START".

Ciò per avere la possibilità di controllare il corretto funzionamento di un programma esaminando una istruzione alla volta e verificando se essa ha eseguito il lavoro assegnato.

Si può impedire automaticamente l'azione del pulsante STEP BY STEP mediante una istruzione particolare chiamata INS.

Se dopo una INS si vuole riabilitare l'arresto del programma occorre inserire una istruzione a ciò preposta e chiamata ENS.

Analizzando il programma istruzione per istruzione, se ne può incontrare una parte che in quel momento non interessa controllare poiché per esempio, si é certi che funzioni correttamente in quanto può essere già stata controllata in precedenza.

In tal caso sarebbe una inutile perdita di tempo continuare a svolgere il programma istruzione per istruzione dando degli START da console. Per ovviare a questo, si può introdurre nel programma, subito prima della parte che si vuole controllare STEP BY STEP, l'istruzione INS sopra esposta.

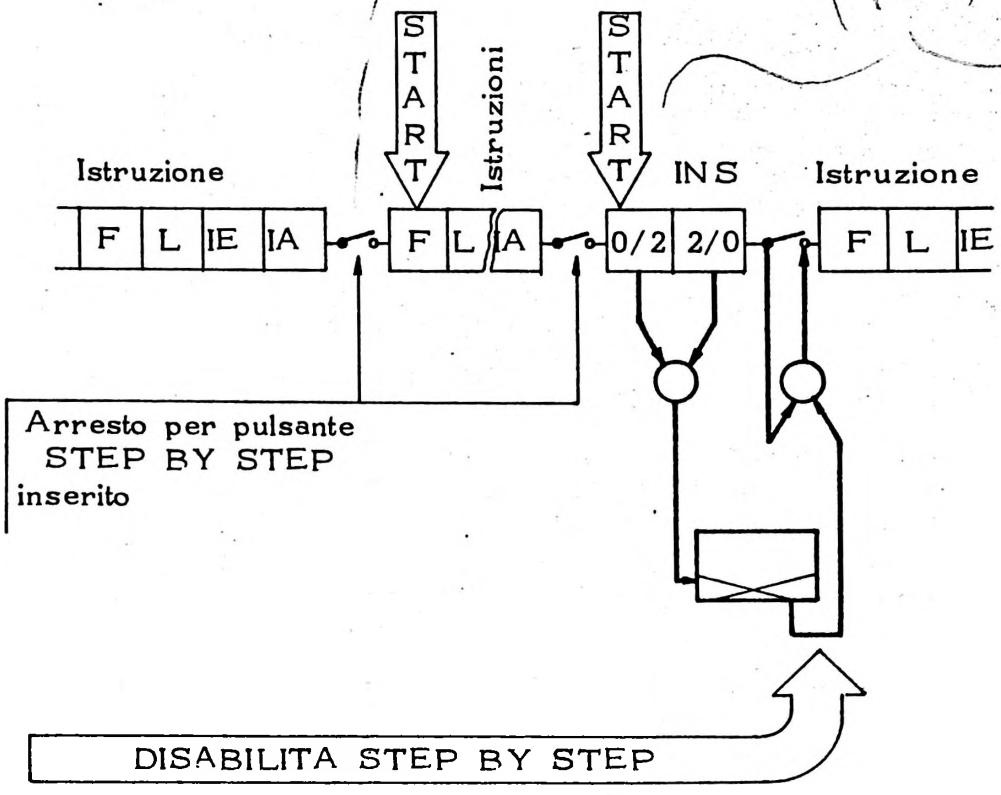
Quando l'elaboratore incontra tale istruzione e la esegue, si verifica che l'effetto del pulsante STEP BY STEP viene annullato ed il programma prosegue automaticamente fino a che non incontra l'istruzione ENS.

15.2. DISABILITAZIONE DEL PULSANTE "STEP BY STEP"

INS (Inhibit Single Stop).

0/2 Codice F' 2/0

Disabilita il pulsante di console "STEP BY STEP".



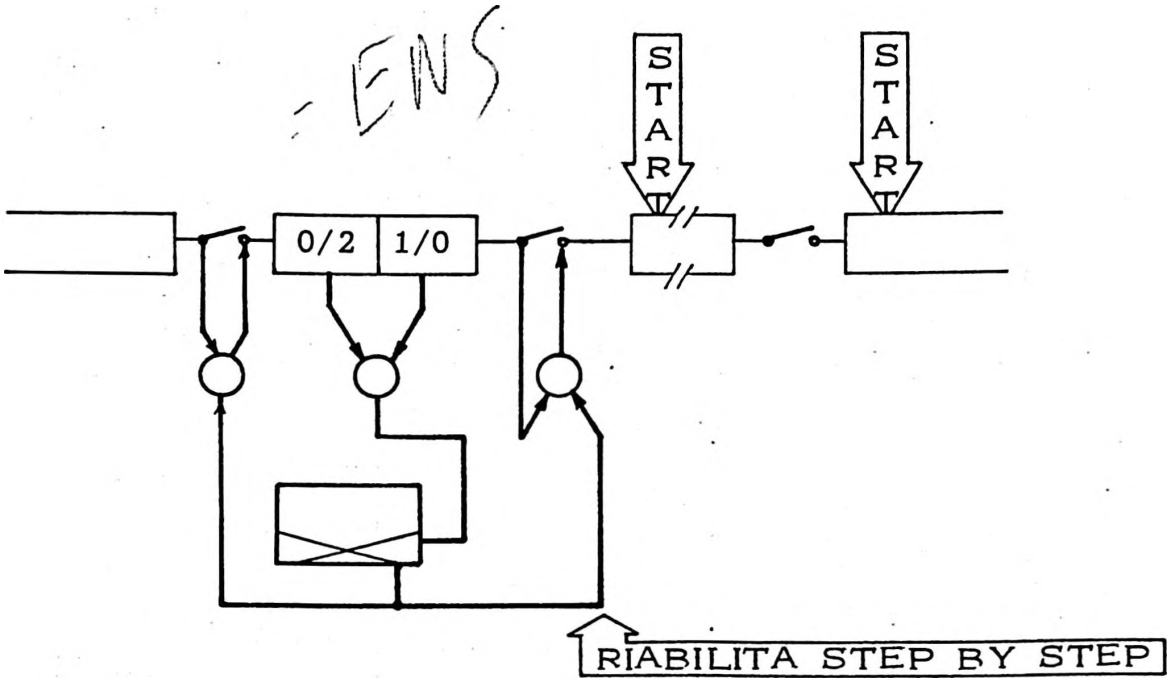
Abilitazione

15.3. ISTRUZIONE DEL PULSANTE "STEP BY STEP"

ENS (Enable Single Stop).

Codice F' 1/0

Abilita il pulsante di console "STEP BY STEP".



DOMANDE

a) - Quali sono le quattro istruzioni del gruppo D4 che hanno lo stesso codice F1 ? INS-ENS LON-LOFF

b) - L'istruzione NOP 2 cosa esegue ? Nessuna operazione

c) - Con quale istruzione si può arrestare l'elaboratore da programma ? HLT o step-by-step ??

RISPOSTE

a) - La INS, ENS, LOFF, LON.

b) - Nessuna operazione.

c) - Con l'istruzione HLT.

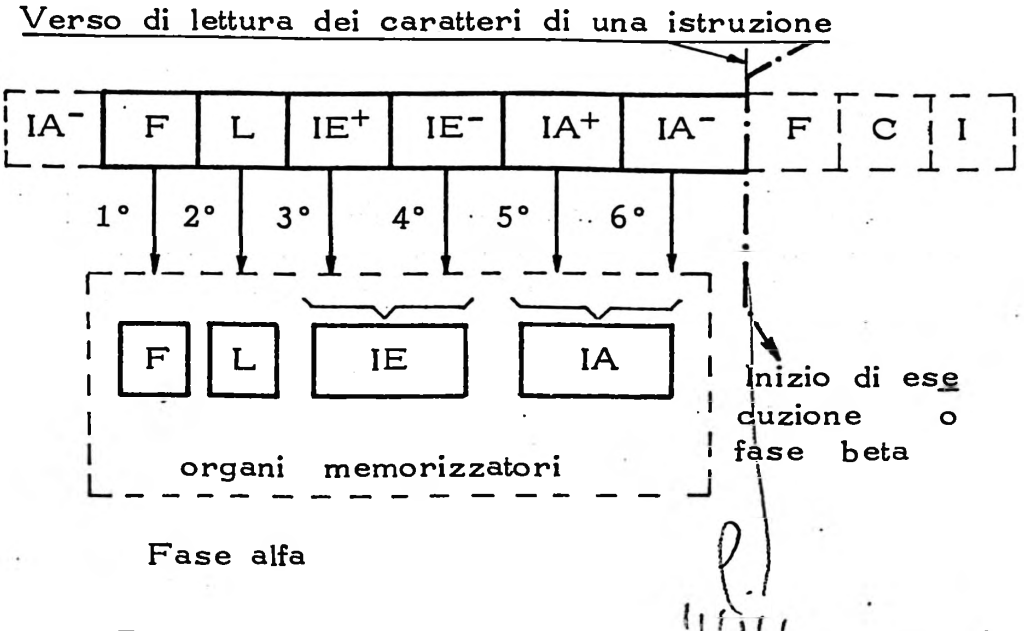
LEZIONE 16

Tempo di studio
1 ora

16. ANALISI DELLE ISTRUZIONI

16.1. FASE ALFA E FASE BETA

Il procedimento di analisi dei caratteri di una istruzione, seguito dall'elaboratore, é uguale per qualsiasi tipo di istruzione interna.



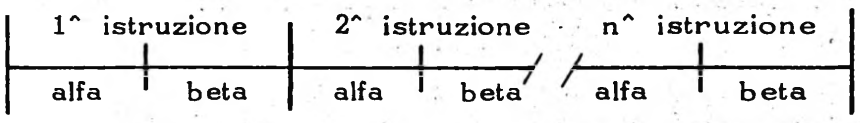
E' necessario sottolineare che l'elaboratore, prima di far eseguire ad un'istruzione l'operazione corrispondente, deve compiere una fase preliminare, chiamata fase alfa o di preparazione, atta a predisporre gli organi interessati all'esecuzione.

La fase alfa é la fase di interpretazione dei caratteri di una istruzione e di predisposizione degli organi di macchina interessati dall'elaborazione successiva (fase beta).

I vari caratteri che compongono un'istruzione assumono, per l'elaboratore il significato di "ordini".

Questi"ordini" devono essere ricordati per tutto il tempo in cui si svolge la fase di esecuzione, o fase beta (se l'istruzione ha fase beta).

Le fasi alfa e beta si susseguono una dopo l'altra.



Al termine di una fase alfa, si inizia l'eventuale fase beta corrispondente, eseguita quest'ultima si avvia la fase alfa dell'istruzione seguente e al termine di questa la eventuale fase beta corrispondente.

Lo svolgimento del programma procede così fino a che non sia comandato all'elaboratore, da programma o con condizioni esterne di console, di arrestarsi.

DOMANDE

a) - Cos'é la fase alfa ? fase di lettura dei caratteri
della istruzione e preparazione di quella per
le fasi successive

b) - Cos'é la fase beta ? la fase di esecuzione dell'atto letto in α.

c) - Come procede l'elaboratore nello svolgimento completo di una istruzione ?
interpreta i caratteri dell'istruzione bel passo gli
ordini e passa a eseguire l'operazione

d) - Come si susseguono le fasi alfa e beta ?

α - β - α β - α ecc

RISPOSTE

a) - E' la fase di interpretazione dei caratteri di un'istruzione e di predisposizione degli organi interessati all'elaborazione.

b) - E' la fase di esecuzione dell'operazione indicata dall'istruzione.

c) - Interpreta i caratteri dell'istruzione, predispone gli organi interessati ed esegue l'operazione.

d) - Ogni istruzione, eseguita la fase di preparazione (alfa), inizia la fase di esecuzione beta. Dopo la fase beta si inizia la fase alfa dell'istruzione seguente e così di seguito fino ad un arresto.

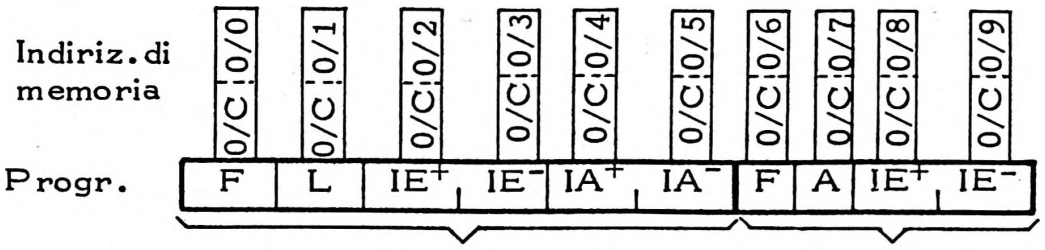
16.2. ORGANO DI INDIRIZZAMENTO PROGRAMMA

Il programma deve essere introdotto, a partire da un certo indirizzo, per indirizzi crescenti.

Le istruzioni che compongono un programma si trovano quindi in memoria a partire da una data posizione.

Esso viene poi letto, istruzione per istruzione, per indirizzi crescenti.

Ciò significa che le istruzioni si trovano sempre predisposte nel seguente modo :



Per l'analisi dell'istruzione é necessario un organo che contenga l'indirizzo iniziale del programma e che possa incrementare il suo contenuto di una unità ogni volta che una posizione di memoria é stata interessata.

DOMANDA Cioè l'indirizzo iniziale

Quale dovrà essere tale organo ?

.....

.....

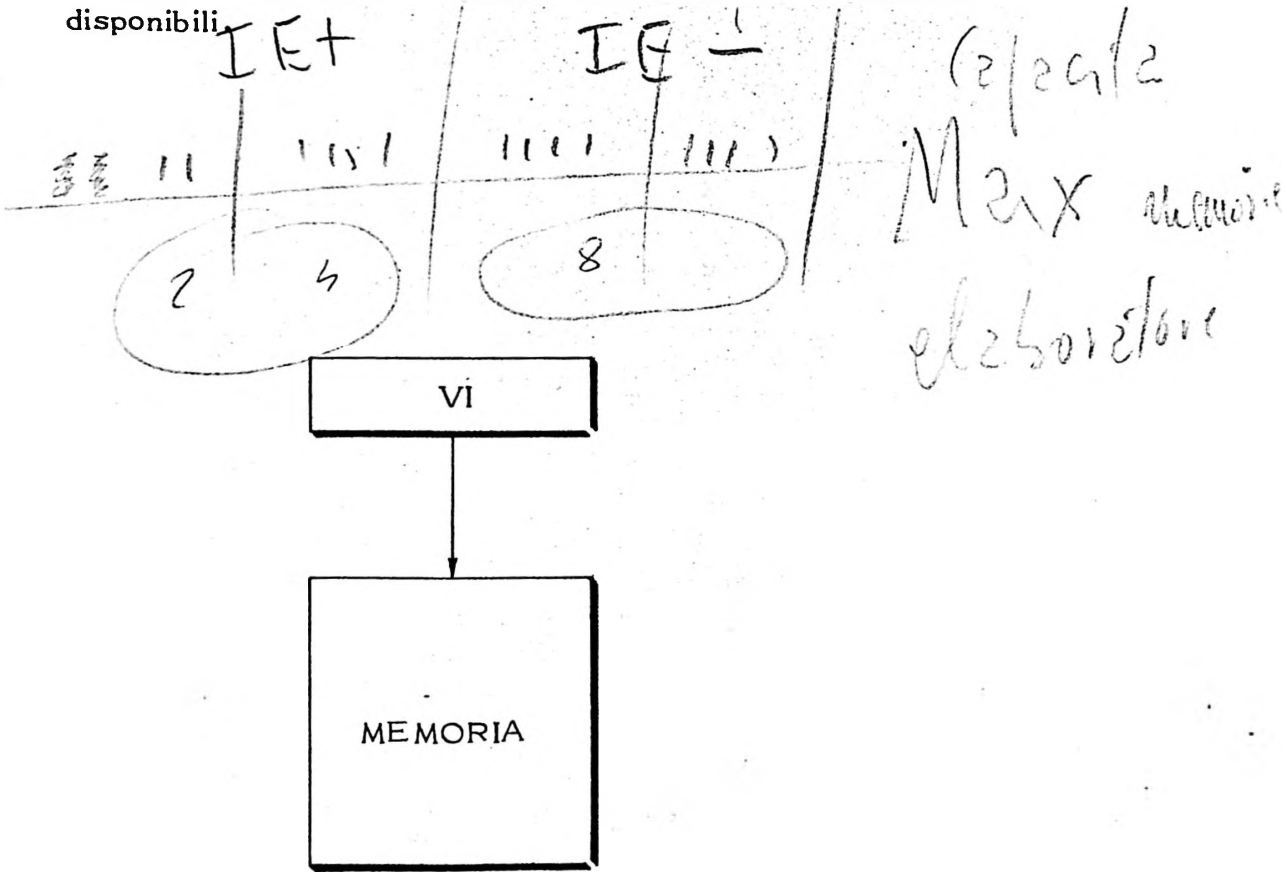
Registro indirizzatore VI.

VI

RISPOSTA

L' organo che contiene le posizioni di programma e che progredisce sempre per indirizzi crescenti é un registro indirizzatore che si chiama VI.

Esso é composto da un minimo di 12 flip-flop ad un massimo di 14 in relazione alle posizioni di memoria disponibili.



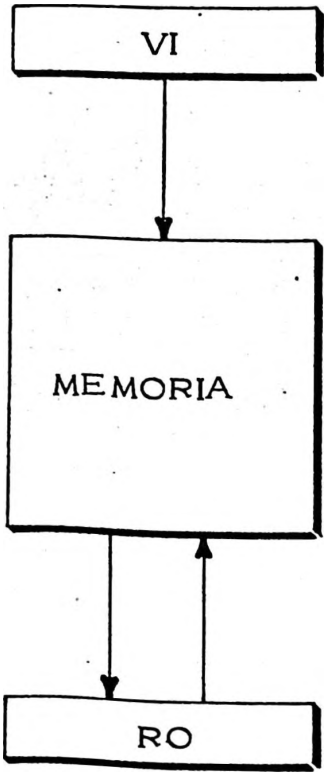
16.3. REGISTRO DI MEMORIA

E' indispensabile prima di proseguire l'analisi degli organi, stabilire che l'elaboratore può leggere un solo carattere alla volta di quelli che si trovano in memoria.

Ogni carattere letto, prima di interessare l'organo prestabilito, viene sempre messo in un registro detto RO. Questo registro é il registro di memoria e riceve sempre, uno alla volta, tutti i caratteri che vengono letti.

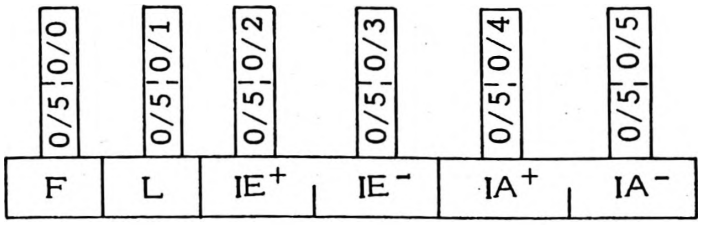
OGNI CARATTERE LETTO DURANTE LA FASE DI PREPARAZIONE VIENE SEMPRE RISCRITTO NELLA STESSA POSIZIONE DI MEMORIA.

A ciò contribuisce, come si vedrà, il registro RO.



16.4. REGISTRO DI FUNZIONE

Si abbia in memoria a partire dall'indirizzo 0/5 0/0 un programma la cui istruzione sia del tipo A2.



L'indirizzatore VI contiene l'indirizzo 0/5 0/0.

L'elaboratore andrà a leggere nella posizione di memoria corrispondente ed estrarrà F ponendo in RO.

Tale carattere, come tutti gli altri durante la lettura dell'istruzione, verrà riscritto nella stessa posizione e quindi mantenuto inalterato.

IL CARATTERE LETTO, CHE RAPPRESENTA IL CARATTERE DI FUNZIONE, DOVRA' RICORDARE ALL'ELABORATORE CHE TIPO DI ISTRUZIONE STA SVOLGENDO. E' NECESSARIO QUINDI PORLO IN UN DATO REGISTRO E MANTENERVELO FINCHE' NON VERRA' LETTO UN ALTRO CARATTERE DI FUNZIONE.

DOMANDA

Come si chiamerà tale organo ?

.....

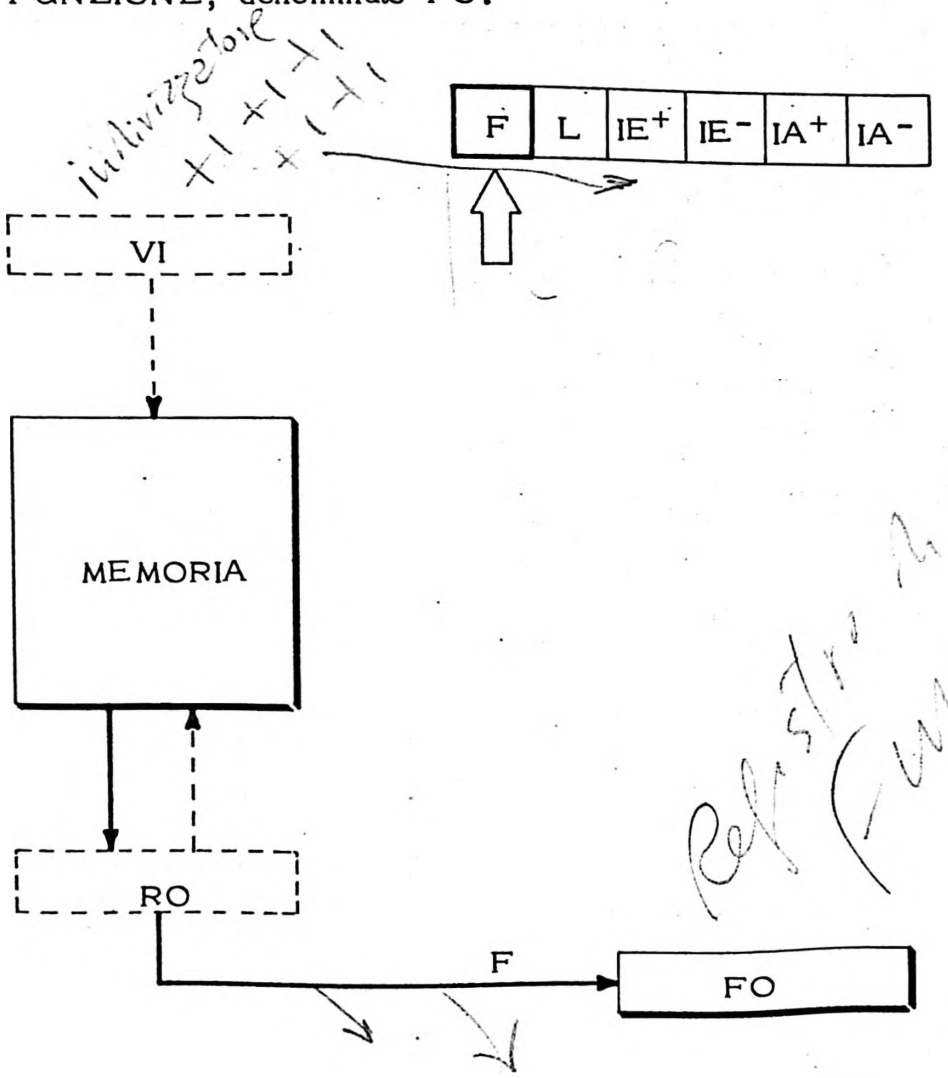
.....

Vedi retro

Registro di Funzione FO

RISPOSTA

L'organo che staticizza per tutta la fase alfa e la fase beta il carattere di funzione é il REGISTRO DI FUNZIONE, denominato FO.



IL 1° CARATTERE LETTO IN FASE ALFA VA IN FO.

16.5. REGISTRO DI LUNGHEZZA

All'indirizzatore VI che conteneva l'indirizzo 0/5 0/0 dopo la lettura del primo carattere F. viene fatto modificare il suo contenuto portandolo a 0/5 0/1, posizione di memoria nella quale si trova il carattere di lunghezza .

Anche per tale carattere ci deve essere un organo che lo contenga per tutto il tempo dell'operazione.

Si legge tale carattere, ponendolo prima in RO e successivamente in un registro che lo contenga per tutto il tempo necessario alla esecuzione dell'operazione.

DOMANDA

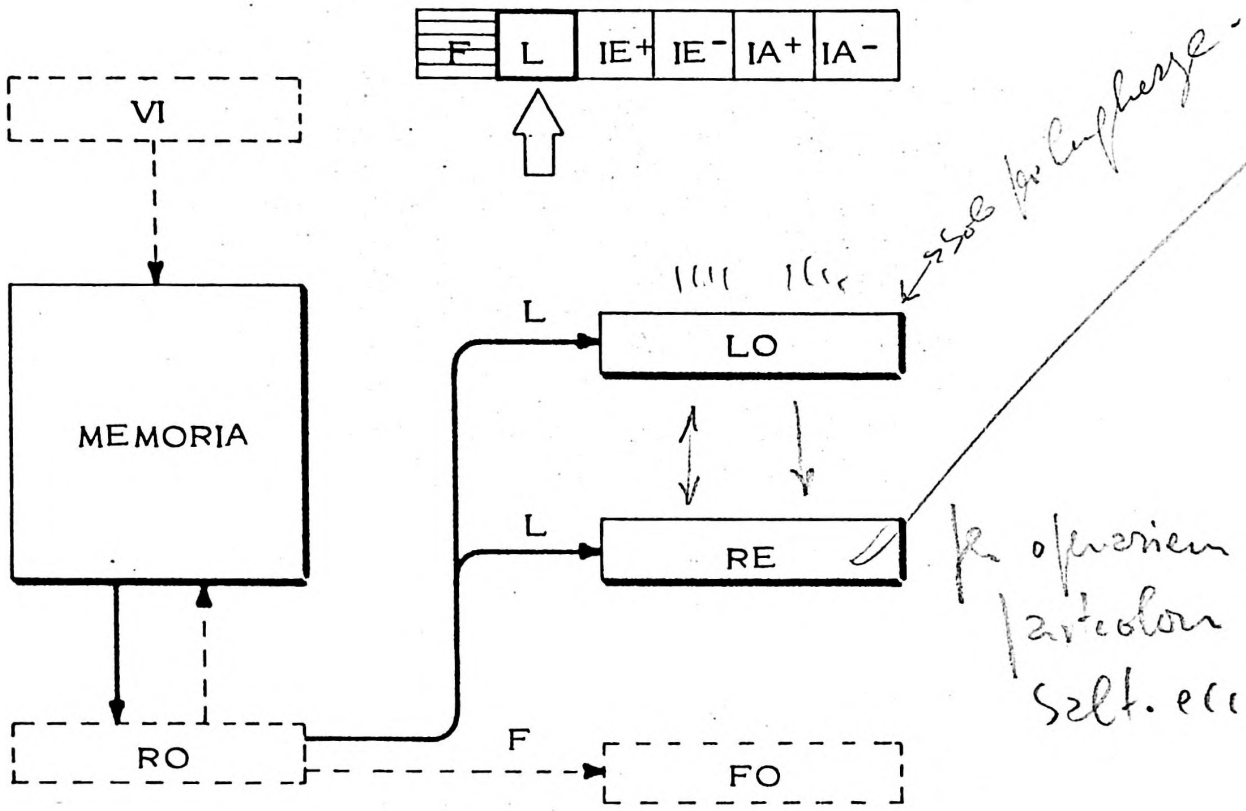
Come si chiamerà il registro che contiene il carattere L ?

.....Registro LO di lunghezza.....

.....

RISPOSTA

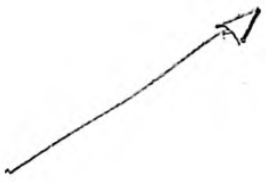
L'organo che contiene il carattere L si chiama REGISTRO DI LUNGHEZZA e viene denominato LO.



IL 2° CARATTERE LETTO IN FASE ALFA VA IN LO.

16.6. IL REGISTRO RE

Il secondo carattere che compone un'istruzione dopo essere stato letto dalla memoria, viene posto oltre che in LO anche in un registro particolare chiamato RE.



L'utilizzazione del contenuto di questo registro si ha durante l'esecuzione di alcune istruzioni particolari (salti, esterne, ecc.) che considerano il secondo carattere che le compongono non come lunghezza bensì come carattere avente uno scopo particolare.

16.7. ORGANI PER L'INDIRIZZAMENTO DEL CAMPO E

Il contenuto di VI continua a progredire di una posizione per ogni carattere letto.

Dopo aver posto il carattere L nel registro LO, l'elaboratore estrae il primo carattere dell'indirizzo iniziale del campo E (IE^+).

IE+ Di tale carattere interessano solamente i primi quattro, cinque o sei bit che rappresentano i bit più significativi dell'indirizzo del campo E. Il numero dei bit che interessano é in relazione al numero di moduli di memoria disponibili (1 ÷ 4).

Questo carattere non esprime quindi l'indirizzo completo del campo, bensì una parte.

IE- Per ottenere l'indirizzo completo del campo é necessario leggere anche il secondo carattere (IE^-).

Il registro RO che contiene il carattere letto dovrà travasare in un altro organo il suo contenuto, prima di poter ricevere il secondo carattere dell'indirizzo del campo E.

Tale organo é il registro RA.

Dopo aver travasato il contenuto di RO (IE^+) in RA l'elaboratore estrae l'altro carattere dell' indirizzo dal campo E (IE^-) e lo pone in RO.

Ora l'intero indirizzo del campo E é estratto dalla memoria, i due caratteri che lo compongono si trovano in RO e in RA.

L'indirizzo completo deve essere ricomposto e portato in un organo capace di memorizzarlo.

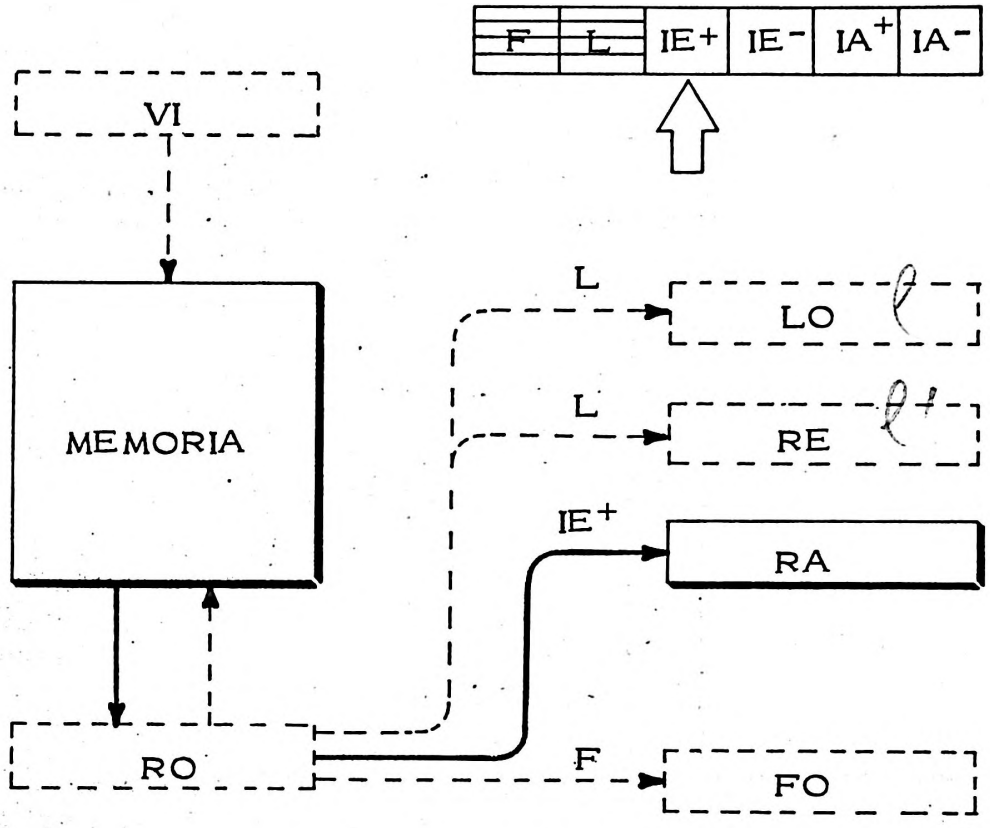
DOMANDA

Quale organo contiene l'indirizzo del campo E ?

.....

Registro mirrogatore del campo E

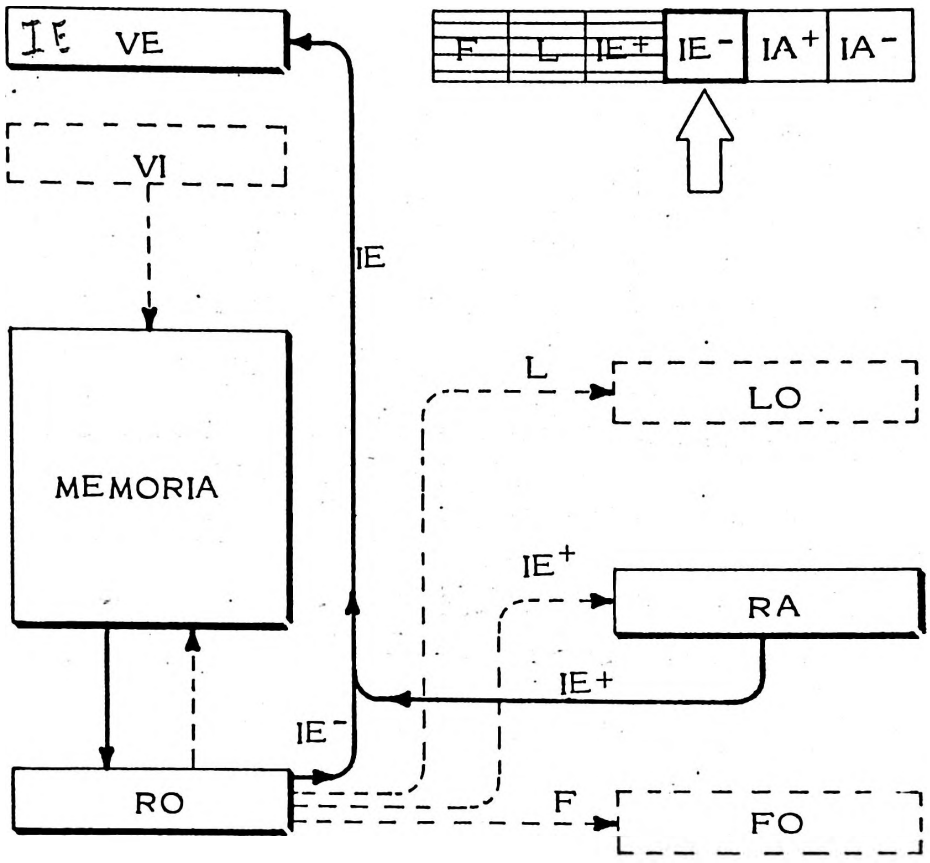
V. E.



IL 3° CARATTERE LETTO IN FASE ALFA VA IN RA.

RISPOSTA

L'organo che memorizza l'indirizzo del campo E é il REGISTRO INDIRIZZATORE DEL CAMPO E che viene denominato VE.

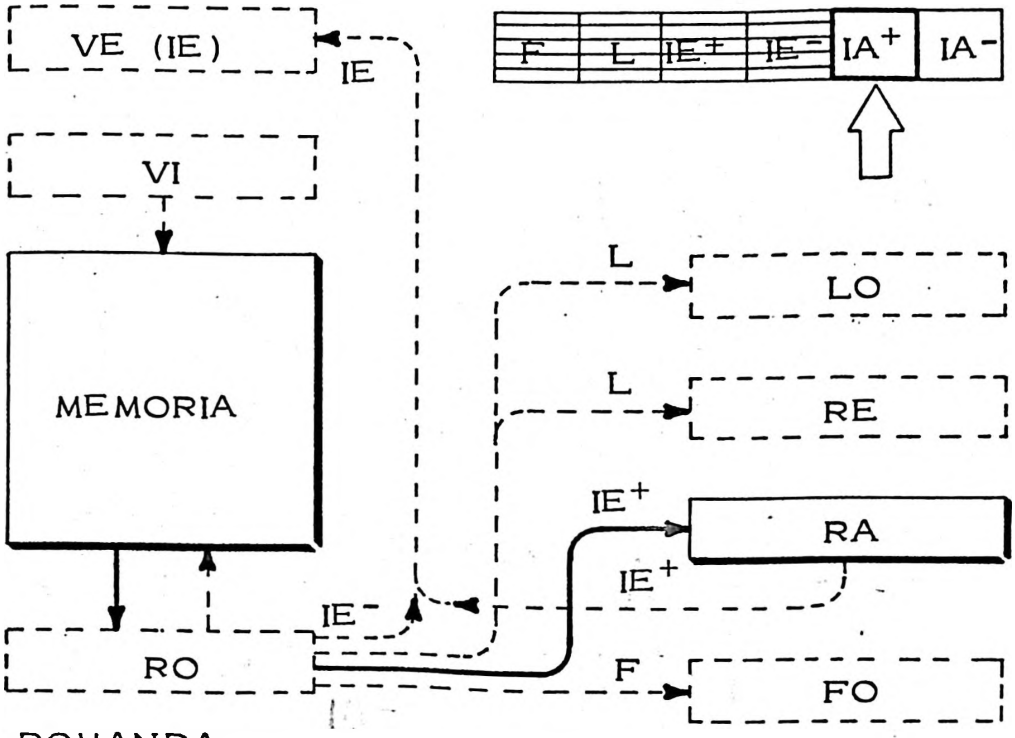


Il 4° CARATTERE RIMANE RO.

RO - RA vengono messi in VE.

16.8. ORGANI PER L'INDIRIZZAMENTO DEL CAMPO A

Come per i caratteri che esprimono l'indirizzo del campo E, così per quelli che esprimono l'indirizzo iniziale del campo A si tratterà di estrarre dalla memoria IA^+ ed IA^- prima di interessare l'organo atto a ricever l'indirizzo completo del campo A.



DOMANDA

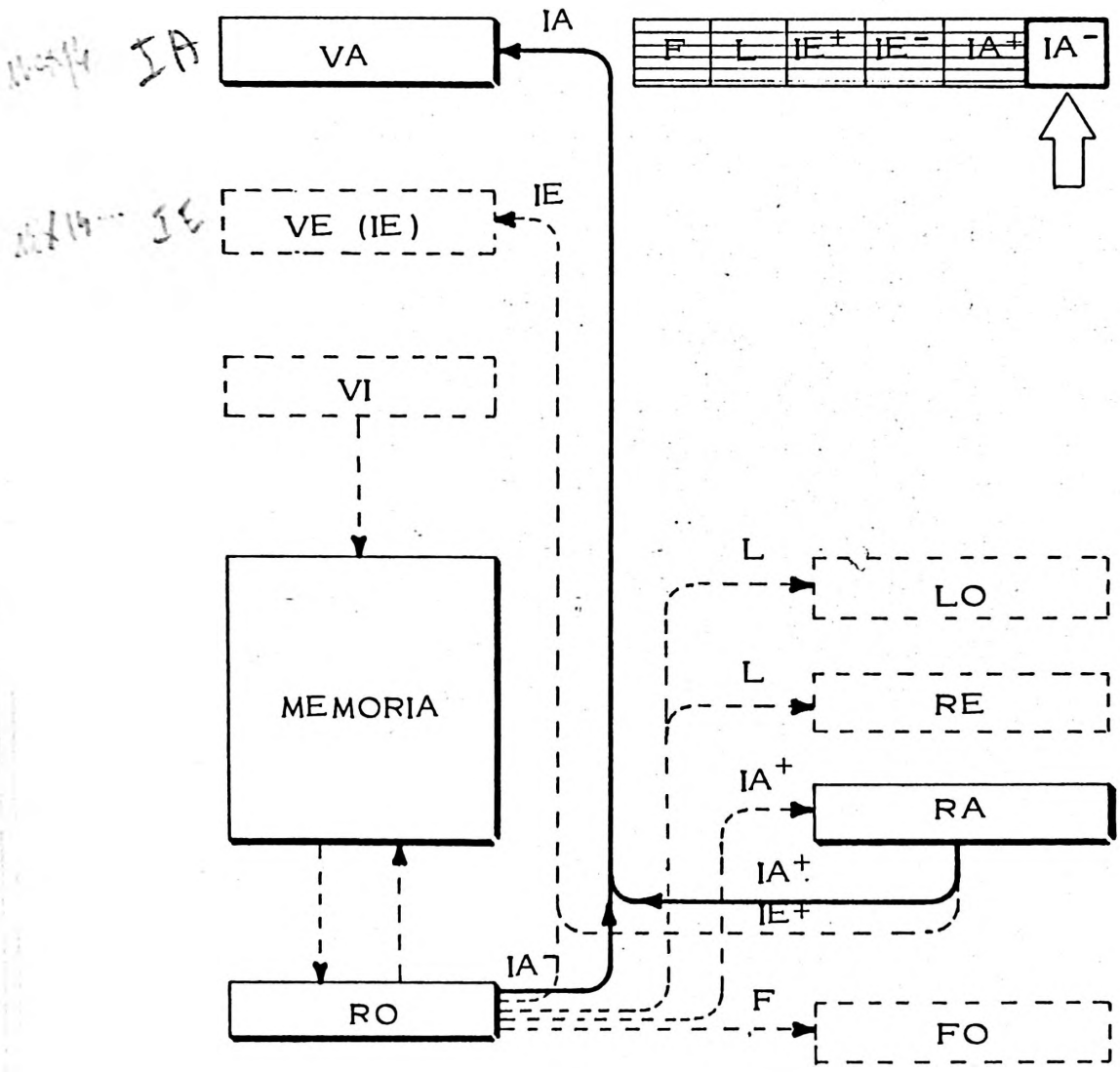
Qual'é l'organo che contiene l'indirizzo completo del campo A?

Reg. Indirizzatore VA

IL 5° CARATTERE LETTO IN FASE ALFA VA IN RA.

RISPOSTA

Esso é il REGISTRO INDIRIZZATORE DEL CAMPO A e viene denominato VA.



IL 6° CARATTERE LETTO RIMANE IN RO.

RO - RA vengono messi in VA.

DOMANDE

a) - Qual'é il registro di funzione e che scopo ha ?

F.O. Il registro F. - contiene la tabella
dei salti.

b) - In fase alfa, che funzione esplica il registro RA ?

registrare IE+ --- IA+ --- le fa vengono insieme
al IE- --- IA- --- le mette nei giusti
registratori VE e VA

c) - Che cos'é RO ?

registratore di ingresso nei circuiti uscenti
dalla memoria microprocessore e diretti ai vari registri:
FO - RE - LO - UE - VA

d) - A che cosa servono VA, VE e VI ?

a registrare IA, IE, e VI affinché +1 ---
+1 determinano la successione del programma
o cellule Memoria nei circuiti dell' ...

e) - Dove si registra la lunghezza ?

LO

RISPOSTE

a) - Il registro di funzione é il registro FO. Ha lo scopo di contenere per tutta l'istruzione (fase alfa e beta) il carattere di funzione.

b) - RA, in fase alfa contiene il primo carattere degli indirizzi dei campi.

c) - RO é il registro di memoria attraverso il quale passano tutti i caratteri letti.

d) - VA contiene l'indirizzo del campo A.

VE contiene l'indirizzo del campo E

VI contiene l'indirizzo del programma.

e) - La lunghezza viene posta nel registro LO.

LEZIONE 17

Tempo di studio
30 minuti

17. FINE FASE ALFA — TEMPORIZZAZIONE

17.1. FINE DELLA FASE ALFA

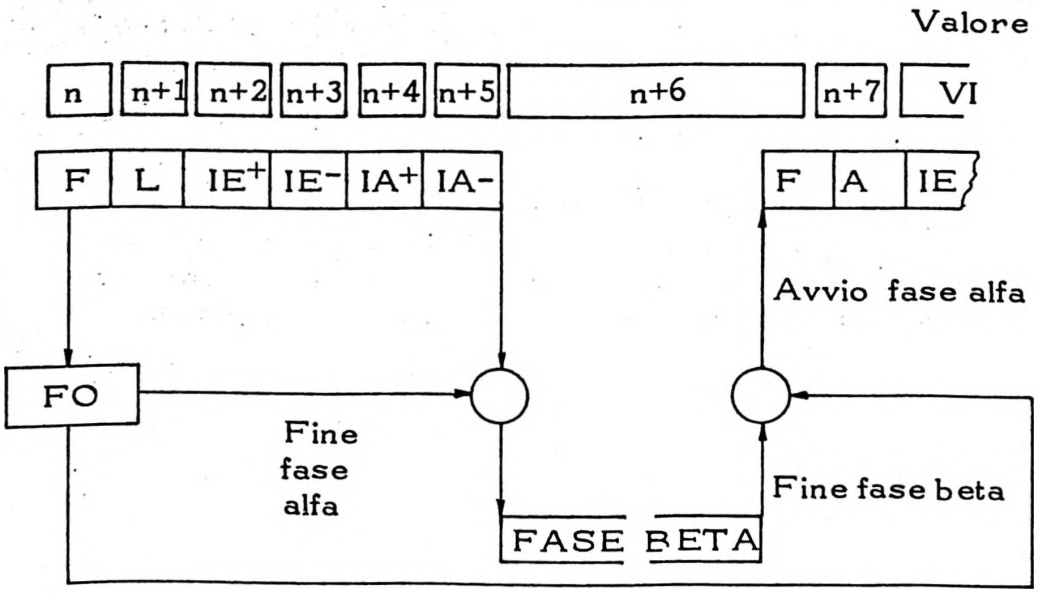
Dopo la lettura del 6° carattere l'indirizzatore VI si predispone per leggere il carattere seguente.

Se ciò avvenisse si leggerebbe il settimo carattere che é il carattere di funzione dell'istruzione seguente (supposto che il programma continui).

Si sa però che prima di iniziare una nuova fase alfa si deve eseguire la fase beta dell'istruzione letta.

Quindi dopo la lettura dell'ultimo carattere dell'istruzione l'elaboratore deve essere avviato alla fase esecutiva.

Ciò viene stabilito tramite il codice di funzione il quale, agendo opportunamente sugli organi interni, fa terminare la fase alfa ed iniziare la fase beta.



Durante lo svolgimento della fase esecutiva si devono eseguire tutte le operazioni elementari che l'istruzione letta prevede.

Per esempio nel caso della MVC trasferire un carattere o più caratteri dal campo A al campo E usufruendo del contenuto di VA e VE.

Durante lo svolgimento della fase beta la memoria deve essere indirizzata ora da VA ora da VE mentre l'ultimo contenuto di VI deve rimanere inalterato per poter ricordare da quale posizione si dovrà riprendere il programma.

L'indirizzatore VA deve quindi indicare alla memoria le posizioni dalle quali prelevare i caratteri, mentre l'indirizzatore VE deve indicare in quali posizioni porli.

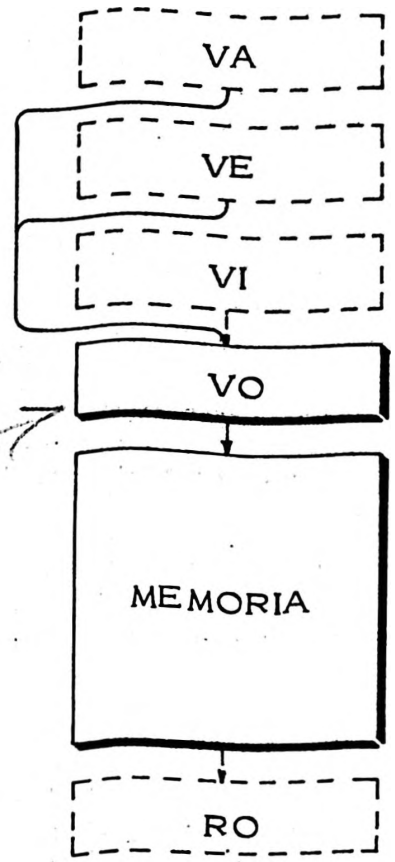
Il programma fornisce ai due indirizzatori le posizioni iniziali dei due campi.

Nel caso in cui i campi siano composti da più caratteri gli indirizzatori dovranno evolvere (come é avvenuto per VI durante la fase alfa) di una unità alla volta per ogni carattere elaborato.

Dato che la memoria deve essere indirizzata sia da VI che da VA e VE, per semplificare la logica dell'elaboratore, é stato introdotto un quarto registro indirizzatore (chiamato VO) che é l'unico che indirizza realmente la memoria e riceve, di volta in volta, il contenuto degli altri tre registri.

Negli schemi precedentemente visti va quindi introdotto VO come in fig.17.1.1.

Quindi VA - VE - VI non indirizzano direttamente con il loro contenuto la memoria ma questo viene travasato in VO che é l'unico indirizzatore a diretto contatto con i circuiti di selezione delle posizioni di memoria.



Seleziona la posiz. di Memoria

fig. 17.1.1

DOMANDE

a) - Chi indica all'elaboratore la lunghezza della fase alfa ? il contatore di lunghezza F? cioè il contenuto di F-0

b) - Che funzione svolge VO ? interfaccia fra VA-VE-VI e i dati circuiti di selezione nella Memoria

c) - Il contenuto di VA, VE, VI dove si pone ?
in V.O.

d) - Cosa dovrà rimanere in VI alla fine della fase alfa ? l'indirizzo da dove ritornerà il programma dopo la fase B.
cioè l'indirizzo del carattere di fusione nell istruzione seguente

RISPOSTE

a) - Il contenuto di FO cioé il carattere di funzione.

b) - VO é l'indirizzatore di memoria che pilota i circuiti di selezione.

c) - Il contenuto di VA, VE, VI si pone sempre in VO.

d) - L'indirizzo del carattere di funzione dell'istruzione seguente.

17.2. CONCETTO DI CONTA DEGLI INDIRIZZATORI

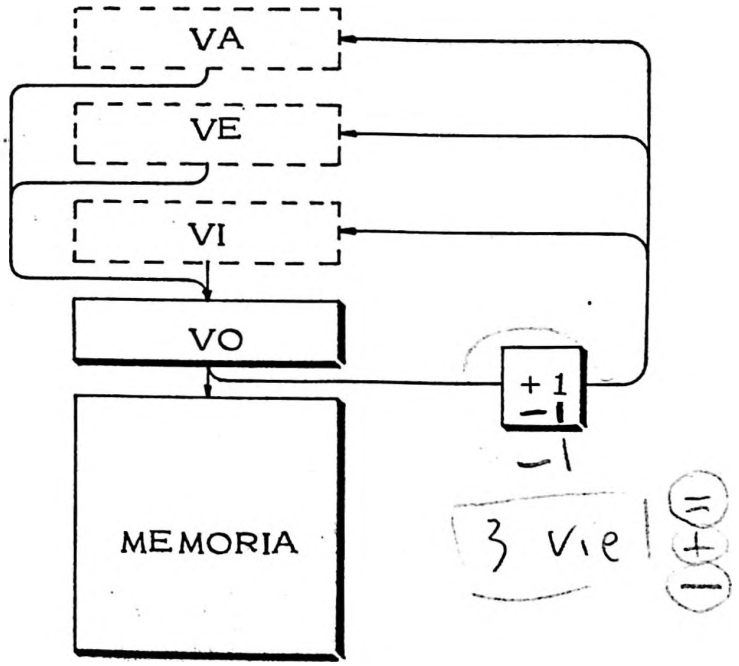
Durante tutta la fase alfa si é considerato che ad ogni carattere letto VI progredisce di una unità.

Il procedimento é il seguente :
VI prima si scarica in VO, poi il contenuto di VO viene rimesso in VI, facendolo passare per una rete di conta.

La rete di conta, che si analizzerà in seguito, provvede a modificare il contenuto di VO non solo per VI, ma anche per VE e VA, prima di riscriverlo nel registro dal quale l'ha ricevuto.

Poiché durante le fasi di esecuzione di alcune istruzioni gli indirizzi dei campi possono anche decrescere, la rete di conta dovrà essere in grado di aggiungere o togliere 1 al contenuto di VO prima di riscriverlo nell'indirizzatore dal quale é pervenuto (VA, VE, VI).

Lo schema di principio si presenterà in questo modo :



Quindi gli indirizzatori VA, VE, VI per poter interessare la memoria dovranno scaricarsi in VO.

L'indirizzatore VO può ricaricare l'indirizzatore dal quale ha ricevuto il contenuto procedendo in tre modi diversi :

- lasciandolo inalterato
- incrementandolo di una unità
- decrementandolo di una unità.

DOMANDE

a) - In quanti modi può agire la rete di conta degli indirizzatori ?

in tre modi +1 / -1 / inalterato.

b) - Dove si trova la rete di conta ?

Fra VO e le entrate in vestibolare di VA-VE-VI

c) - VO ricarica direttamente l'indirizzatore dal quale ha ricevuto il contenuto ?

Sì, nei tre modi suesposti cioè tramite rete di conteggio

RISPOSTE

a) - In tre modi diversi :

- lasciando il contenuto inalterato
- incrementando il contenuto di 1 unità
- decrementando il contenuto di 1 unità.

b) - Tra l'indirizzatore VO e gli indirizzatori VA, VE e VI.

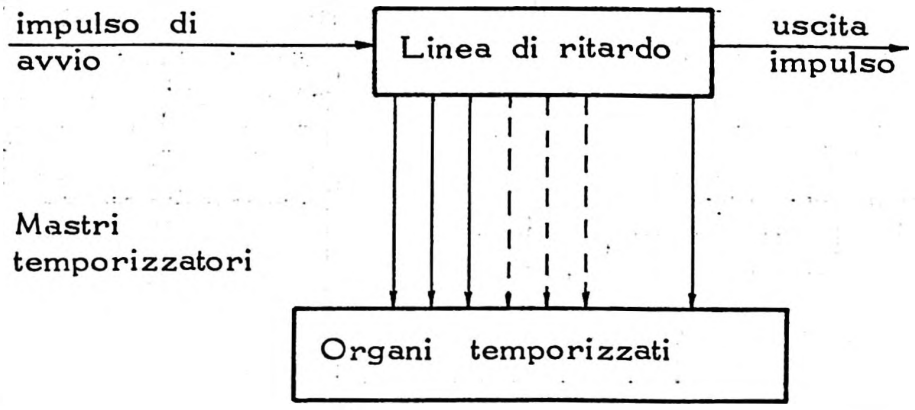
c) - No, il suo contenuto deve prima essere interessato dalla rete di conta.

17.3. TEMPORIZZAZIONE

Come si é potuto costatare nell'esposizione delle precedenti lezioni, l'estrazione dei dati della memoria viene eseguita in tempi successivi.

L'organo temporizzatore, cioé l'elemento che scandisce gli eventi successivi di lettura e riscrittura in memoria e di altre operazioni elementari, é la linea di ritardo.

In questo organo entra un impulso che viene ripreso in punti diversi ad intervalli di tempo diversi.



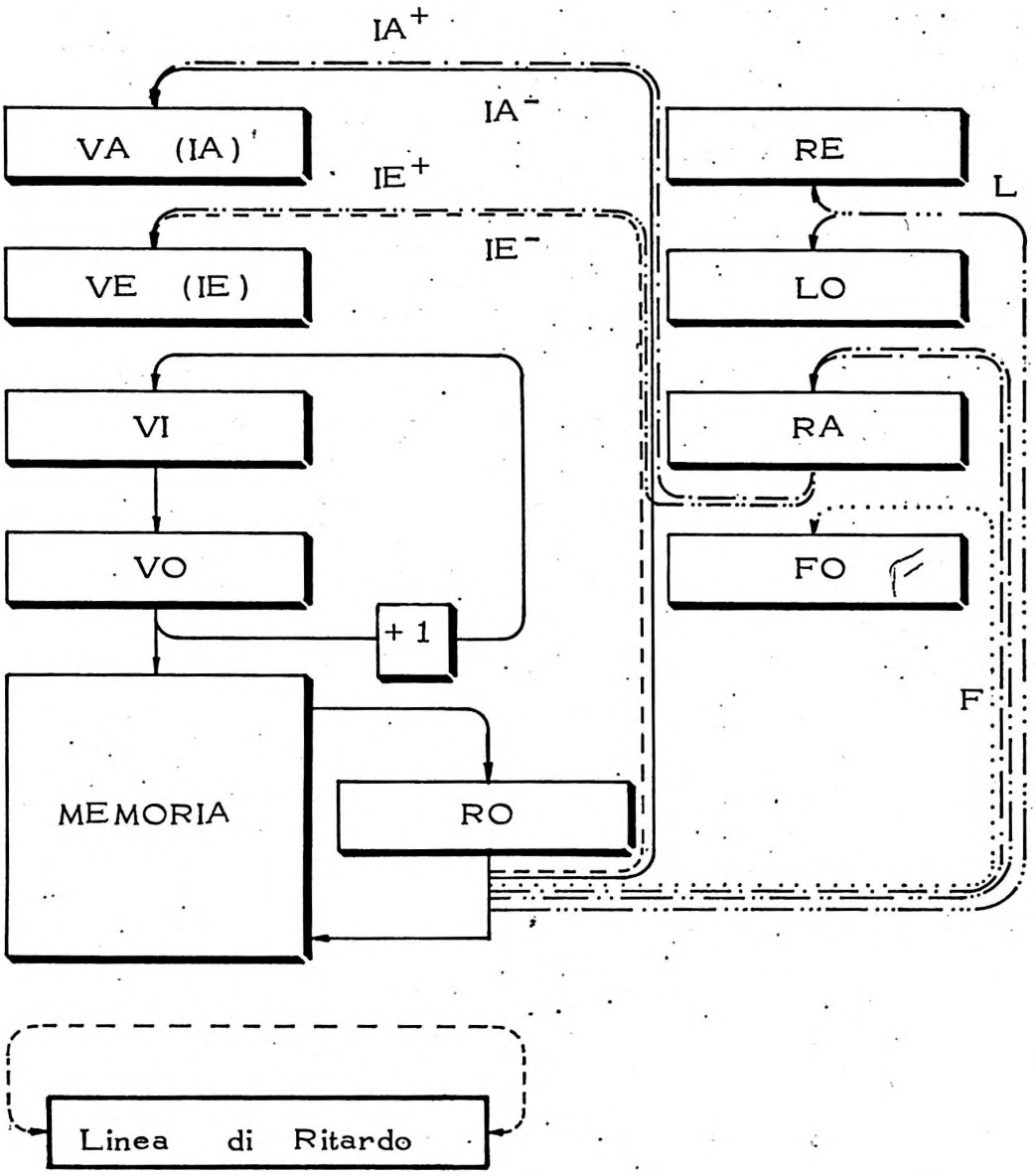
L'intervallo totale di tempo, tra l'ingresso e l'uscita dell'impulso può essere o di 6 o di 7,5 /usec.

DURANTE QUESTO INTERVALLO SI ESEGUE SEMPRE LA LETTURA E SCRITTURA DI UN CARATTERE IN MEMORIA.

I segnali intermedi servono per temporizzare gli eventi di macchina che interessano una "ciclata" di linea di ritardo.

Ora si può fare una esposizione generale degli organi che vengono interessati dall'elaboratore durante la fase di preparazione di una istruzione.

SCHEMA DEGLI ORGANI INTERESSATI NELLA FASE DI PREPARAZIONE DI UNA ISTRUZIONE



·····1° Crt. F ·—·—3° Crt. IE^+ ·—·—5° Crt. IA^+

·——·—2° Crt. L ------4° Crt. IE^- ———6° Crt. IA^-

DOMANDE

a) - Al termine della fase alfa cosa si trova nei registri FO - LO - VE - VA - VI ?

F L IE IA ↳ l'indirizzo nel
contatore di fase della prima istruzione

b) - Ha importanza il contenuto di RO e RA alla fine della fase alfa ?

NO

c) - Dopo la fase alfa quale altra fase deve iniziare ?

B

RISPOSTE

a) - In FO c'é il carattere di funzione della istruzione letta.

In LO la lunghezza dell'operando sul quale si dovrà agire.

In VE l'indirizzo iniziale del campo E.

In VA l'indirizzo iniziale del campo A.

In VI l'indirizzo del carattere di funzione dell'istruzione seguente.

b) - No. Infatti questi organi sono serviti solo come registri ausiliari durante questa fase.

c) - La fase di esecuzione o fase beta. Cioé la vera e propria operazione indicata dall'istruzione letta, ammesso che la istruzione lo richieda.

LEZIONE 18

Tempo di studio
2 ore

18. SVOLGIMENTO DI UNA ISTRUZIONE

18.1. LA MATRICE LOGICA E GLI "STATI"

Terminata la fase di predispossizione degli organi, l'elaboratore deve iniziare la vera e propria operazione che l'istruzione ha il compito di svolgere. L'elaboratore può eseguire un'operazione alla volta in relazione all'istruzione analizzata.

Sia la fase di preparazione che quella di esecuzione dell'operazione é frazionata nel tempo; cioé si arriverà al termine dell'operazione per eventi successivi. Un insieme di eventi viene chiamato STATO.

La successione di STATI é sempre uguale per l'esecuzione della stessa operazione e per determinate condizioni.

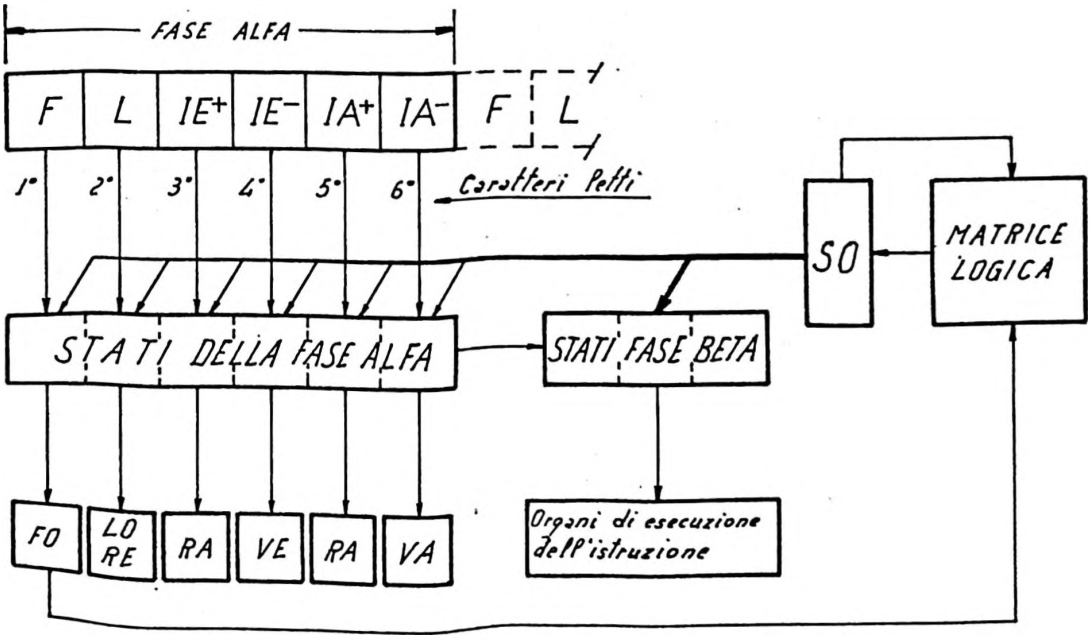


fig. 18.1.1

Gli STATI sono diversi gli uni dagli altri e la loro sequenzialità é stabilita in maniera rigorosa.

L'organo che predispone la via al tipo di esecuzione e che stabilisce il susseguirsi degli STATI é la MATRICE LOCICA.

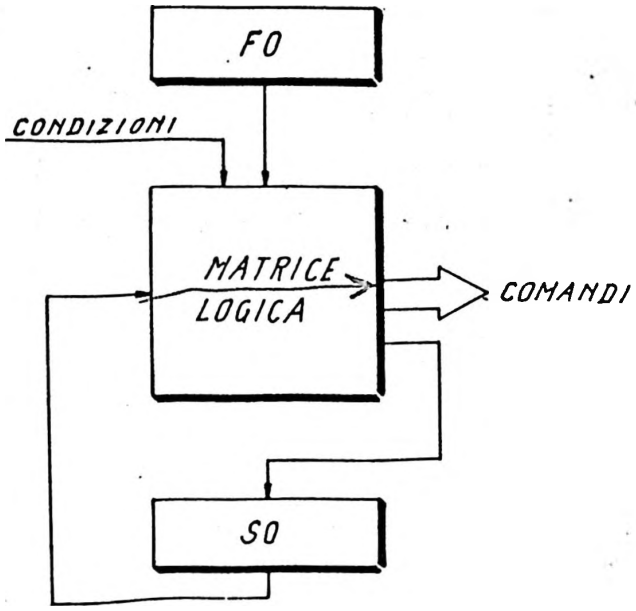
Nell'istruzione compare un codice differenziatore di tutte le operazioni che é il codice di funzione.

Quest'ultimo é quindi un elemento che deve influenzare la sequenza degli STATI della matrice logica.

Infatti al variare di F dovrà variare la via che si vuole seguire poiché ad ogni F corrisponde una operazione diversa.

Il codice di funzione é solo uno degli elementi che condiziona la matrice logica.

La matrice logica, per stabilire uno STATO, deve tener conto non solo del condizionamento dato dal codice di funzione, ma anche delle condizioni che gli STATI precedenti hanno generato.
Quanto esposto può essere schematizzato in fig.18.1.2.



fig; 18.1.2

Uno stato durante il suo svolgimento, in relazione alle condizioni presenti in macchina e al codice di funzione, stabilisce anche l'uscita dello STATO successivo.

In definitiva si può genericamente stabilire che la matrice logica dà origine a degli insiemi di eventi condizionati a tre fattori :

- il contenuto di FO (elemento distintivo delle istruzioni)
- lo STATO precedente
- le condizioni stabilite nel calcolatore dagli STATI precedenti.

Ogni singolo STATO é identificato da più COMANDI che sono generati dalla matrice logica.

Uno STATO rimane presente per il tempo di una ciclata di L.d.R.

L'ORGANO CHE DEFINISCE LA CONFICURAZIONE DECLI STATI SI CHIAMA REGISTRO SO.

DOMANDE

a) - Qual'é l'organo che stabilisce il susseguirsi degli STATI ?

La Matrice logica

b) - Cos'é uno STATO ?

Quella istruzione di comando che le cond. (F, vol. Costr. precedenti e istruzioni codificate.

c) - Per quanto tempo rimane presente uno STATO ?

Una ciclata

RISPOSTE

a) - La matrice logica.

b) - Uno STATO é un insieme di comandi che determina una frazione dell'operazione.

c) - Per il tempo di una ciclata di linea di ritardo.

18.2. CONTA IN LO

Nello svolgimento della fase esecutiva si elaborano i dati. L'elaborazione prosegue finché tutti i dati non sono stati interessati.

L'elemento che stabilisce la quantità di caratteri sui quali operare é la lunghezza degli operandi o dell'operando che si indica nella istruzione.

I caratteri vengono elaborati uno alla volta.

Per ogni carattere (o coppia di caratteri) elaborato é necessario che un organo ricordi l'avvenuta elaborazione e che, al termine di questa, segnali la fine dell'esecuzione.

L'organo adibito a tale scopo é il registro LO il quale deconta di una unità per ogni carattere (o coppia di caratteri) elaborato.

Per poter eseguire la DECONTA in LO si avrà una rete di conta adibita a tale funzione.

Il registro LO e la relativa rete di conta é schematizzato in fig. 18.2.1.

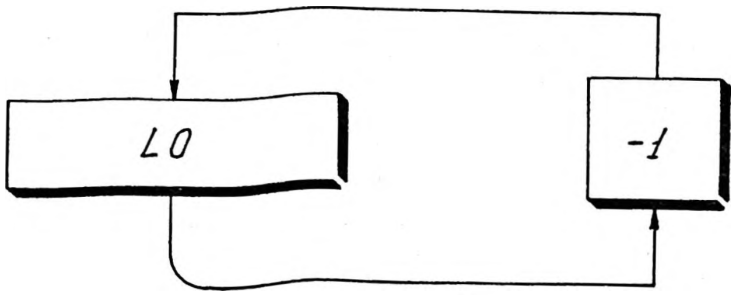


fig. 18.2.1

Quando il registro LO avrà raggiunto una data configurazione (tutti uno), si avrà la fine dell'istruzione come si può vedere nello schema di fig. 18.2.2.

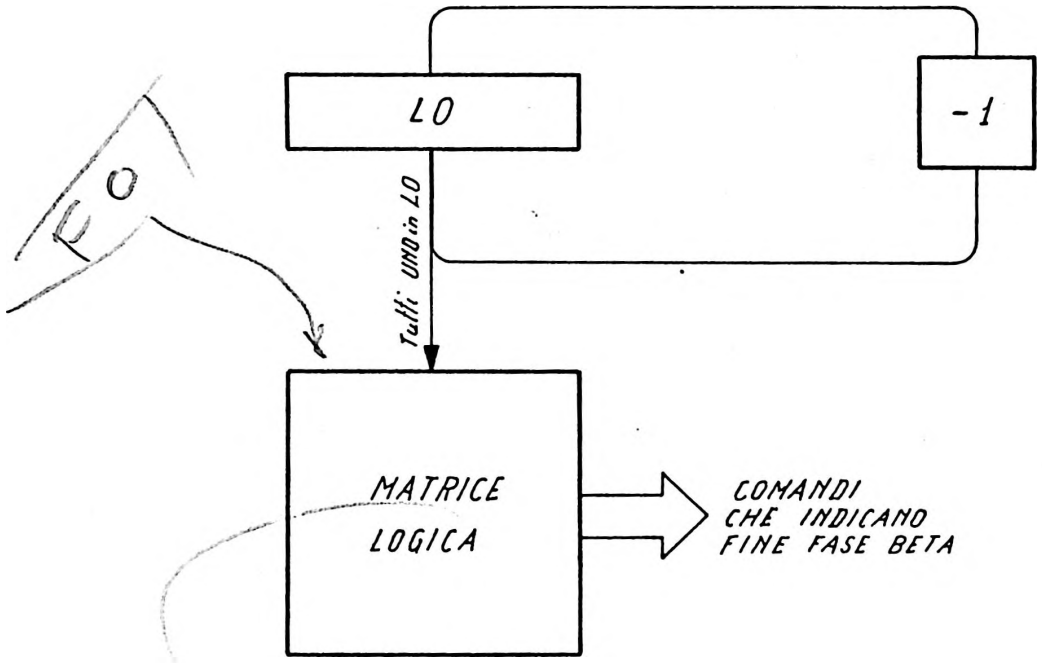


fig. 18.2.2

DOMANDE

a) - Perché LO deve decontare ?

Sulle L = Mezzo elaborare + 1 / el. da sommare 1

Solo contare alla volta ... 1 - 1 - 1 = 0 - 1 = 1111

b) - Quando si avrà la fine dell'elaborazione ?

Ora LO sarà pieno di 1111

RISPOSTE

a) - LO deconta per ricordare l'avvenuta elaborazione di un carattere o di una coppia di caratteri.

b) - Quando, per deconte successive, LO si é svuotato e assume la configurazione prestabilita di tutti uno (00000000 - 1 = 11111111).

18.3. CONTA O DECONTA DEGLI INDIRIZZATORI

Al termine della fase di preparazione, gli indirizzatori dei campi contengono l'indirizzo dal quale si inizia l'estrazione del carattere da elaboratore.

A seconda del tipo di istruzione lo svolgimento dell'elaborazione potrà essere o per indirizzi crescenti o per indirizzi decrescenti.

Nel primo caso (indirizzi crescenti) il contenuto degli indirizzatori dovrà progredire, per ogni carattere elaborato, di una unità.

Nel secondo caso (indirizzi decrescenti) il contenuto degli indirizzatori dovrà regredire di una unità.

Ciò é ottenuto agendo con opportuni comandi sulla rete di conta degli indirizzatori.

Se un determinato evento di una istruzione richiede che non venga modificato il contenuto di un registro indirizzatore, l'elaboratore procede nel seguente modo :

- non viene trasferito il contenuto della rete di conta nel registro
- viene inibita la rimessa a zero del registro stesso.

Come conseguenza si ha che il registro mantiene lo stesso indirizzo che aveva precedentemente.

Durante la fase beta gli indirizzatori (VA - VE) interessano VO con il loro contenuto uno per volta su comando della matrice logica.

In fig. 18.3.1 sono riportate le interconnessioni fra i registri indirizzatori, la rete di conta e la memoria.

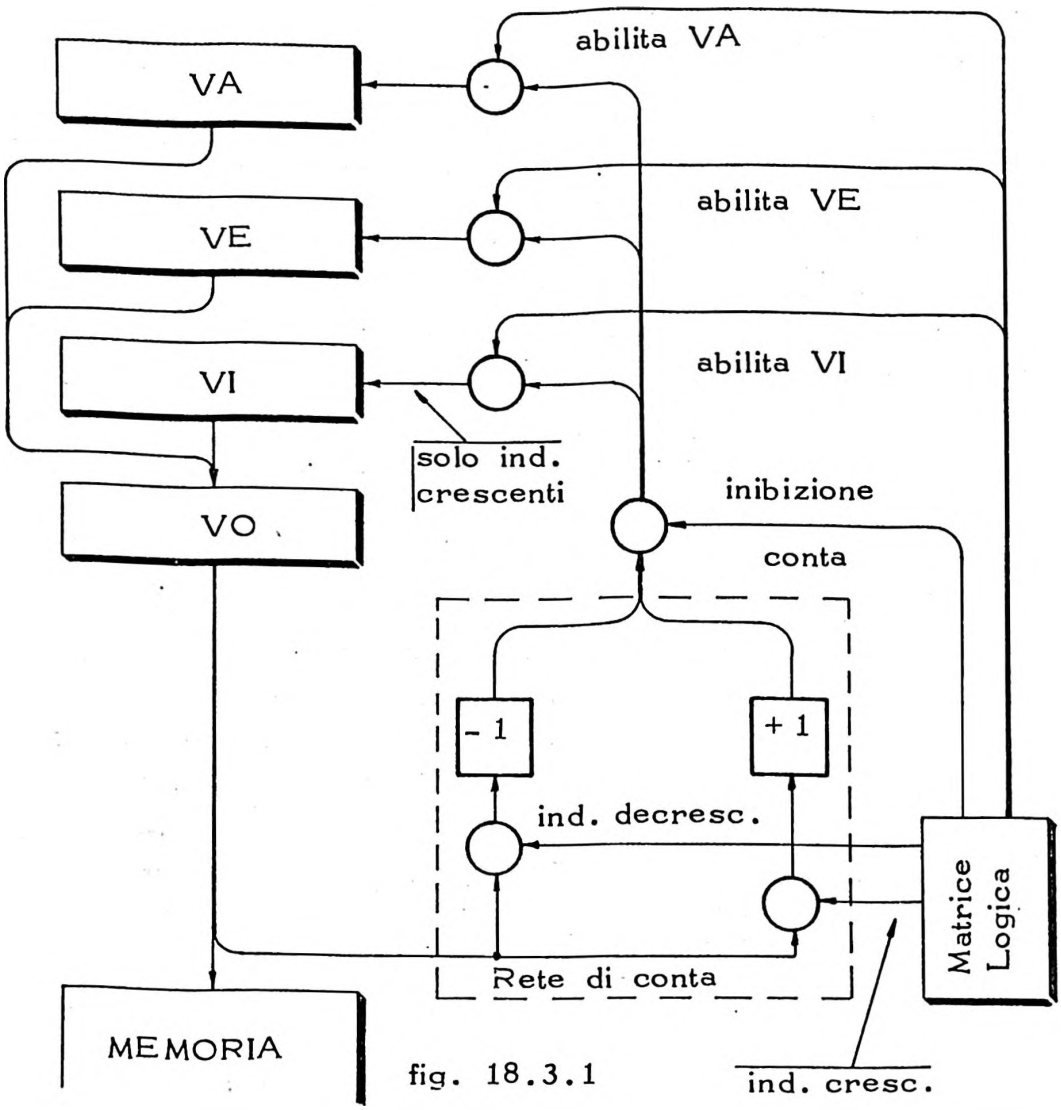


fig. 18.3.1

E' la matrice logica che stabilisce se deve essere VE a scaricarsi in VO oppure deve essere VA.

Questo, é necessario sottolinearlo, si verifica poi ché VO é l'unico indirizzatore di memoria e può contenere solo un indirizzo di selezione per volta.

Durante una fase di elaborazione in cui vi siano due campi, VO verrà interessato alternativamente da VA e poi da VE o viceversa. Per ogni indirizzo che si presenta in VO si farà l'estrazione del carattere da elaborare che si trova in quella posizione di memoria.

Per cicli successivi si estrarranno tutti i caratteri fino a che LO non segnalerà fine esecuzione.

18.4. UNITA' ARITMETICA

L'unità aritmetica é l'organo dal quale si deve ottenere il risultato di una operazione aritmetica decimale o binaria.

Gli ingressi dell'unità aritmetica devono quindi essere interessati dai caratteri sui quali si vuole operare.

Opportuni comandi agiscono in maniera da determinare quale tipo di operazione aritmetica é necessario svolgere.

L'unità aritmetica riceve i dati da RO e da RA.

Il risultato viene scritto in RO cancellando il precedente contenuto.

Si supponga di voler sommare due numeri posti nelle due zone di memoria indicate da VA e VE.

Si deve fare prima un ciclo per poter estrarre un carattere (VA) e porre questo carattere in RA (1° operando).

Poi con un successivo ciclo si estrarrà il secondo carattere (VE) e lo si pone in RO, (2° operando) quando RO é caricato si abiliterà la U.A. ad eseguire l'operazione. Il risultato dell'operazione aritemetica si scrive in RO cancellando il secondo operando, indi da RO in memoria, nella stessa posizione precedentemente occupata dal secondo operando.

Quanto sopra esposto é schematizzato in fig.18.4.1.

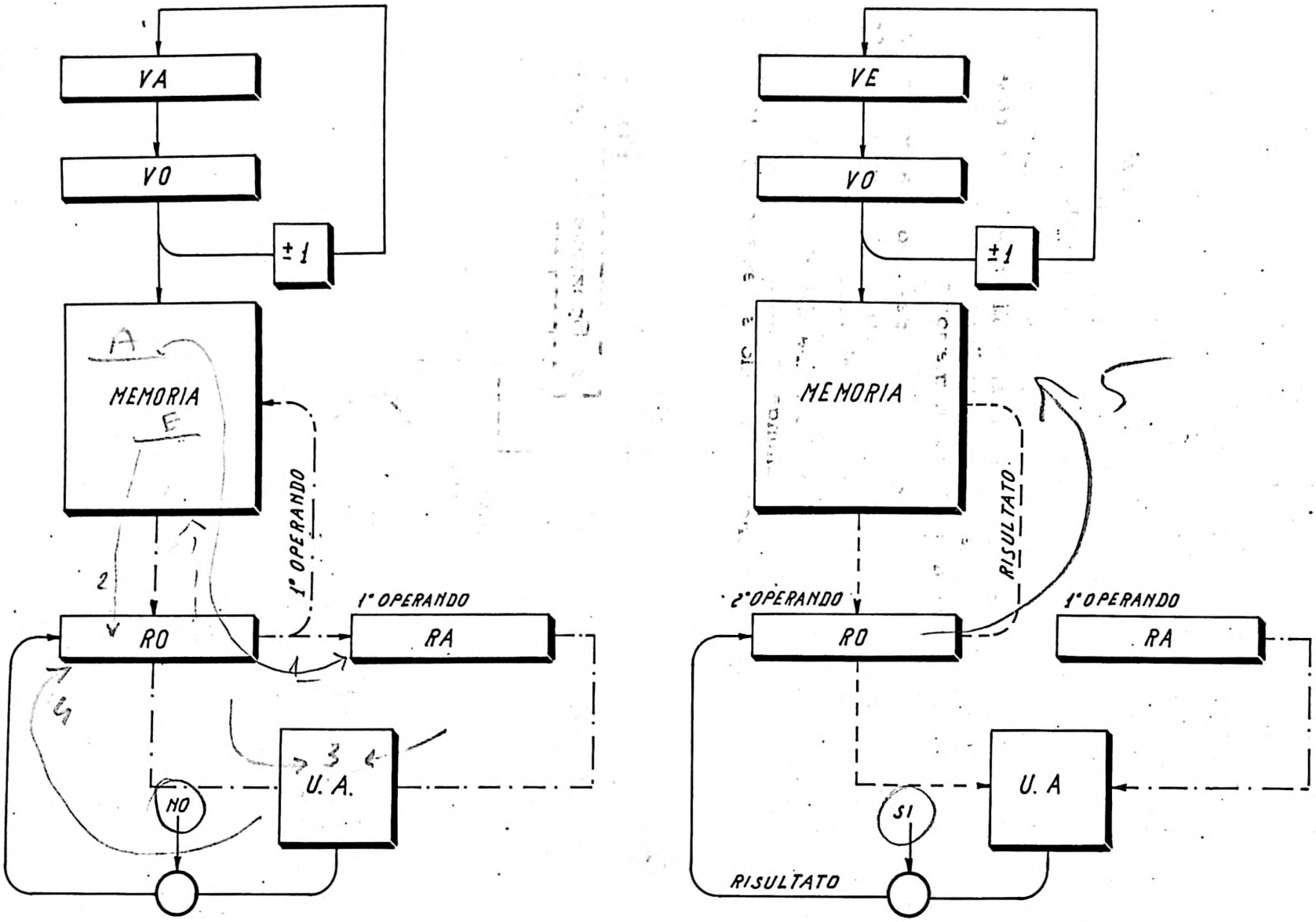

VA
VO
±1
MEMORIA
A
E
1° OPERANDO
RO
RA
U. A.
NO
VE
VO
±1
MEMORIA
RISULTATO
2°OPERANDO
1°OPERANDO
RO
RA
U. A
SI
RISULTATO

fig. 18.4.1

18.5. SCRITTURA DI UN CARATTERE IN MEMORIA

Quando viene utilizzata l'unità aritmetica la scrittura del risultato in memoria non può avvenire nello stesso tempo previsto per scrivere un carattere normale.

Quando agisce l'U.A. il carattere che si deve scrivere non é quello estratto, ma il risultato dell'operazione aritmetica che viene posto in RO.

Dal momento in cui RO riceve il carattere dalla memoria al momento in cui si sostituisce tale carattere con il risultato dell' U.A., la linea di ritardo emette il comando per la scrittura in memoria.

In tale caso, se non si adottassero accorgimenti particolari si scriverebbe nuovamente, in memoria, il carattere estratto invece che il risultato.

Per consentire la scrittura del risultato di una operazione aritmetica si allunga la linea di ritardo di 1,5 usec come rappresentato nello schema di fig. 18.5.1.

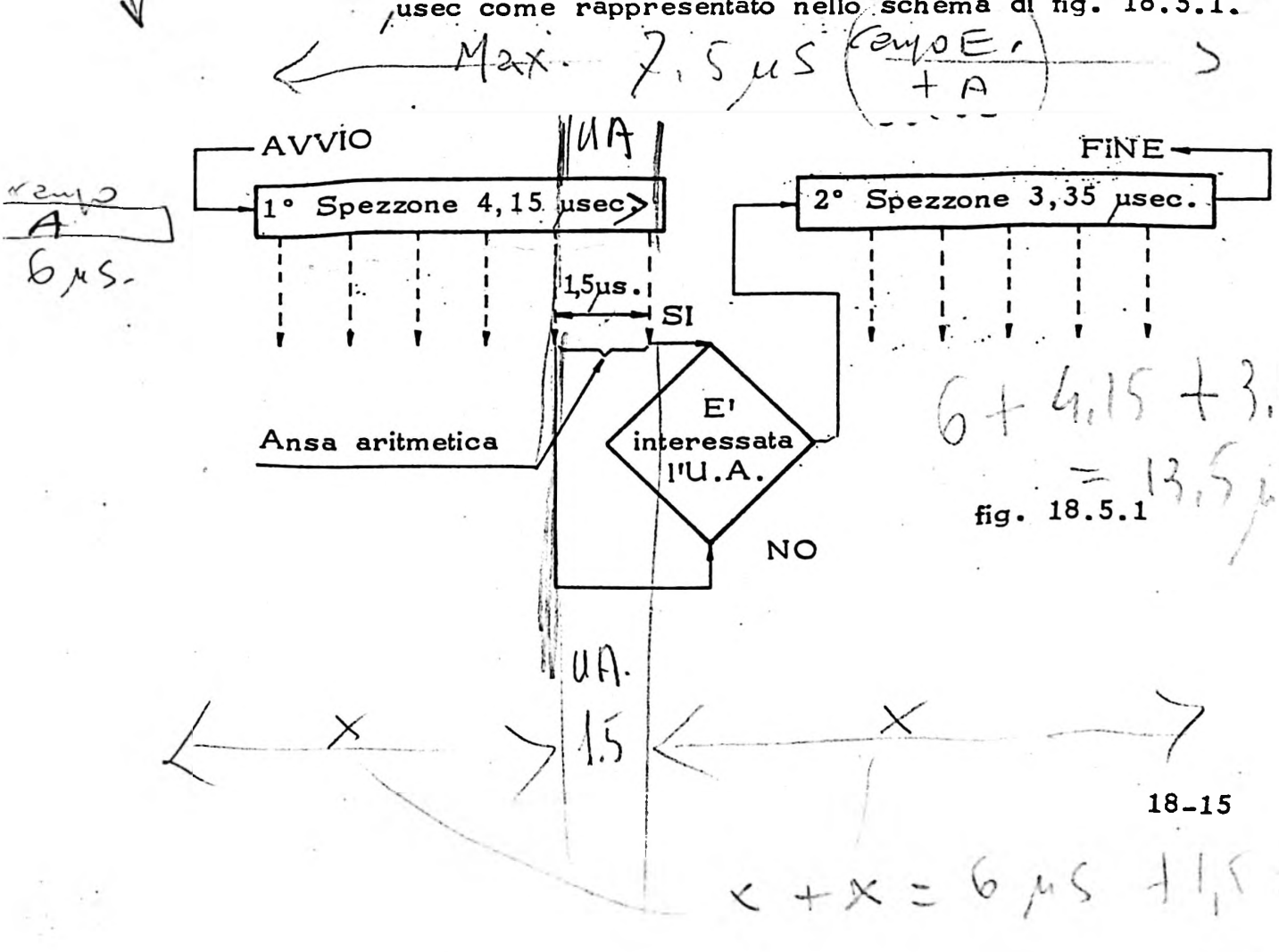


fig. 18.5.1

I mastri del 1° spezzone della L.d.R. temporizzano sempre la lettura da memoria.

I mastri del 2° spezzone della L.d.R. temporizzano sempre la scrittura in memoria.

L'ansa aritmetica viene inserita con un sistema logico quando viene interessata l' U.A.

L'ansa aritmetica serve per far sí che il 2° spezzone sia fatto partire con un ritardo di 1,5 usec rispetto al ciclo normale, in modo da poter caricare RO con il risultato che proviene dall' U.A.

DOMANDE

a) - A cosa serve l'U.A. ?

A sommare o sottrarre un carattere da una altra (Decimale / Binario)

b) - Quali sono i registri che interessano l'U.A. ?

Due - RO / RA

c) - In quante ciclate di L.d.R. si esaurisce una operazione aritmetica relativa a due caratteri ? A+E

Due (at 7 1/5) (6+6+1/5) Risult.

d) - Perché si deve "allungare" la linea di ritardo quando viene interessata la U.A. ?

Si scrive in memoria il dato elaborato dalla UA e non il dato originale estratto...

e) - Dove va il risultato dell'U.A. ?

— RO

f) - In quale posizione di memoria si mette il risultato ?

nella loc. del numero cartaE

(dove si era estratto E I E

RISPOSTE

a) - Serve per eseguire le operazioni binarie o decimali di somma o di sottrazione.

b) - Sono RA e RO.

6 + 2,5

c) - In due ciclate : con la 1^ si pone il primo operando in RA, con la 2^ si pone il secondo operando in RO e si esegue l'operazione.

d) - Per consentire a RO di ricevere correttamente il risultato dell'U.A. prima di scriverlo in memoria.

e) - Va nel registro RO.

f) - Nella posizione stabilita dalla configurazione di VO relativa all'ultimo indirizzatore che si é scaricato (VE). Tale indirizzo é quello relativo al secondo operando.

18.6. ORGANI AUSILIARI

In alcune istruzioni (confronti, ricerche ecc.) si ottengono determinate segnalazioni. E' necessario, quindi al termine della loro fase di esecuzione, mantenere inalterata la segnalazione ottenuta per poterla successivamente analizzare con altre istruzioni. Gli organi addetti a questa funzione, sono i flip-flop RIPO, DIVE, DISA e COEN. Essi mantengono i risultati dei confronti (RIPO e DIVE) e di altre operazioni (aritmetiche, di servizio ecc.).

Ad essi, appunto in relazione alle varie utilizzazioni, si attribuisce il nome di organi ausiliari.

Con le nozioni che si possono ottenere dall'esposizione fin quì fatta si può rappresentare lo schema di principio del calcolatore.

Bisogna tenere presente che anche in fase alfa interviene la matrice logica, ma per semplicità di esposizione ciò non era stato detto.

Inoltre nell'analisi dettagliata delle istruzioni interne (volume II U.C.) si vedrà come taluni organi sono collegati tra loro in maniera da consentire certe operazioni.

Lo schema a blocchi degli organi interessati durante le istruzioni interne é riportato in fig. 18.6.1.

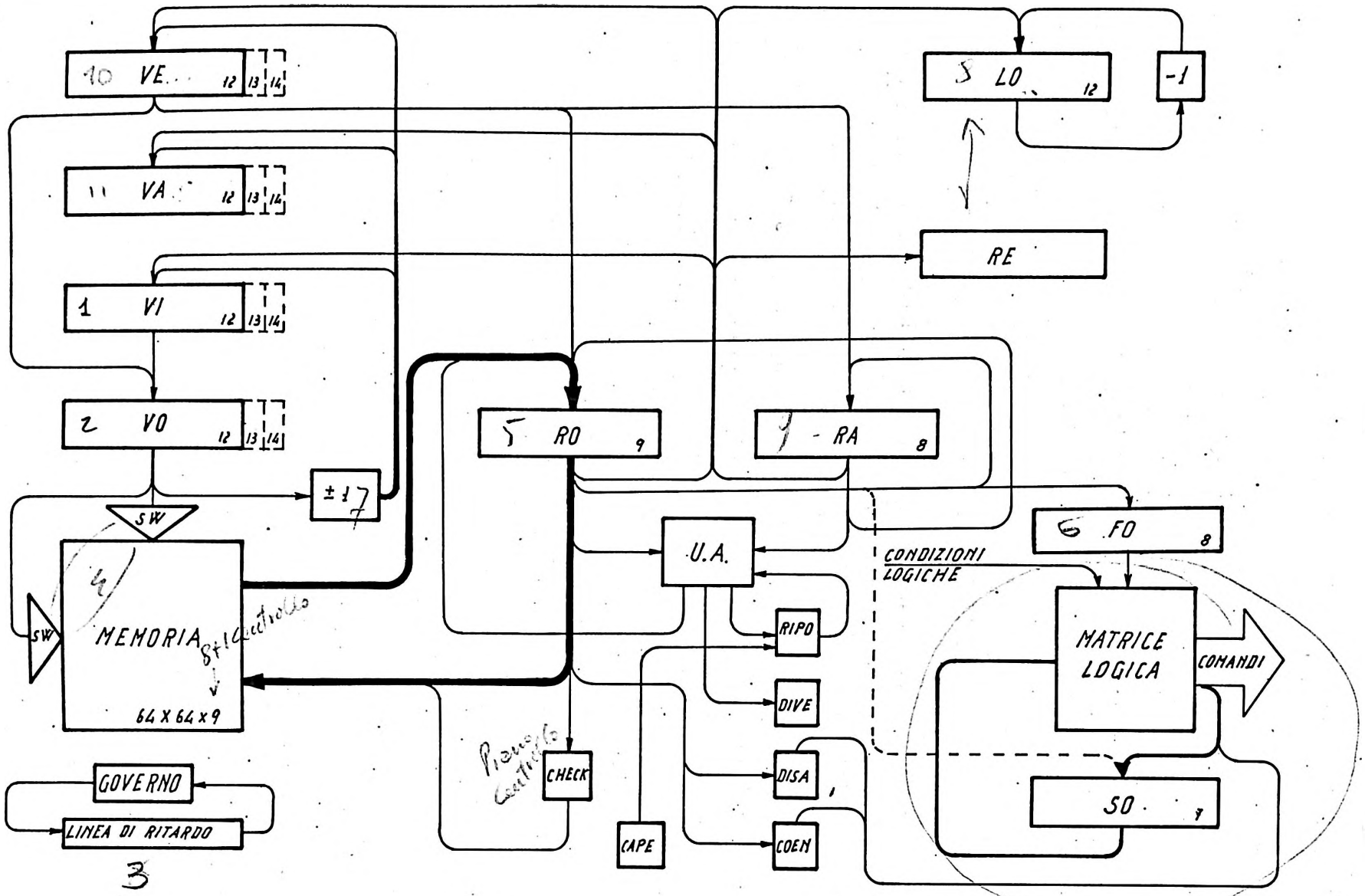

10 VE 12 13 14
11 VA 12 13 14
1 VI 12 13 14
2 VO 12 13 14
5 LO 12
-1
RE
5 RO 9
7 - RA 8
±17
SW
SW
MEMORIA
64 X 64 x 9
U.A.
RIPO
DIVE
DISA
CHECK
CAPE
COEN
CONDIZIONI LOGICHE
6 FO 8
MATRICE LOGICA
COMANDI
SO 9
GOVERNO
LINEA DI RITARDO
3

fig. 18.6.1

DOMANDE

a) - Quali sono gli organi che vengono utilizzati nella fase alfa di un'istruzione di tipo A2 ?

tutti / esclusi MA, RF, Rip, Dv, Dse, Cod, Cpc

b) - Quali sono gli organi utilizzati nella fase beta delle A2 ?

V. 1 e 2 seq.

c) - Che funzione hanno i singoli organi ?

//

d) - Qual'é l'"indirizzatore della fase alfa ?

Vedi

RISPOSTE

a) - In ordine di utilizzazione :

1 - L'indirizzatore VI (indirizzo iniziale istruzione)
2 - l'indirizzatore VO (per ogni indirizzo di memoria)
3 - la linea di ritardo
4 - la memoria e i circuiti di selezione
5 - il registro RO (registro di memoria)
6 - il registro FO (registro di funzione)
7 - la rete di conta indirizzatori
8 - il registro LO (registro lunghezza)
9 - il registro RA (registro di deposito IE^+ e IA^+)
10 - l'indirizzatore VE (inizio campo E tramite RO e RA)
11 - l'indirizzatore VA (inizio campo A tramite RO e RA)

La rete di conta interviene ogni volta che si é interessato un carattere; per semplicità ciò non é stato ripetuto ogni volta.

Unità Aritm. RE (non utilizzato).
Perché non c'è anche Matrice logica. Vd. 18-19

b) - Gli indirizzatori VE e VA e la rete di conta degli indirizzatori.

4 - la memoria e i circuiti di selezione
8 - il registro LO e la sua rete di conta
7 - l'unità aritmetica
x6 - la matrice logica, il registro SO e il registro FO
5-9 - i registri RA e RO
3 - la linea di ritardo
x - i flip-flop RIPO e DIVE
2 - il registro VO.

c) - In ordine alfabetico :

- DIVE F.F. ausiliario
- FO contiene il codice di funzione
- L.d.R. temporizza gli eventi
- LO registro di lunghezza
- matrice logica invio comandi e nascita di eventi
- memoria deposito dati e programma
- RA registro di deposito
- RIPO F.F. ausiliario
- RO registro di memoria
- rete di conta indirizzatori (conta ± 1)
- rete di conta LO (conta - 1)
- SO condiziona la matrice logica
- U.A. opera sui dati aritmeticamente o in binario
- VA contiene gli indirizzi del campo A
- VE contiene gli indirizzi del campo E
- VI contiene gli indirizzi di programma
- VO indirizzatore unico di memoria (pilota i circuiti di selezione memoria).

d) - L'indirizzatore VI.

LEZIONE 19

Tempo di studio
1 ora

19. STUDIO DI UN PROGRAMMA

19.1. REALIZZAZIONE DEL FLOW-CHART

Il programma di un elaboratore é stato definito una successione ordinata di istruzioni.

Utilizzando le istruzioni del GE - 115, esposte nelle precedenti lezioni, si procede allo studio di un programma, dapprima realizzandone il "flow-chart" cioé uno schema a blocchi, quindi specificando i caratteri delle istruzioni.

Si supponga di avere perforato, sulla stessa scheda, i dati :

X che può variare da 1 a 10 cifre, e

Y che può variare da 1 a 8 cifre.

Su un'altra scheda sono perforati i dati :

Z che può variare da 1 a 9 cifre, e

W che può variare da 1 a 8 cifre.

Si vuole eseguire la seguente elaborazione :

- introdurre in memoria i dati X e Y perforati su una scheda,
- introdurre in memoria i dati Z e W perforati su una altra scheda,
- eseguire la somma, in valore decimale, dei dati X e Y ottenendo un totale T (X + Y = T),
- eseguire la sottrazione, in valore decimale, del dato W al dato Z ottenendo il risultato R (Z - W = R),

- stampare i dati introdotti e i risultati ottenuti nel seguente modo :

X Y T Z W R

Per introdurre dei dati, da unità periferiche, in memoria sono necessarie delle opportune istruzioni, denominate "ISTRUZIONI ESTERNE" che saranno esaminate nel volume "U.C. vol. 3°".

Per ora queste istruzioni sono indicate con la loro denominazione e non con i relativi 4 caratteri.

Nel programma in esame sono necessarie due istruzioni esterne di introduzione dei dati, una per ogni scheda (vedasi fig. 19.1.1) :

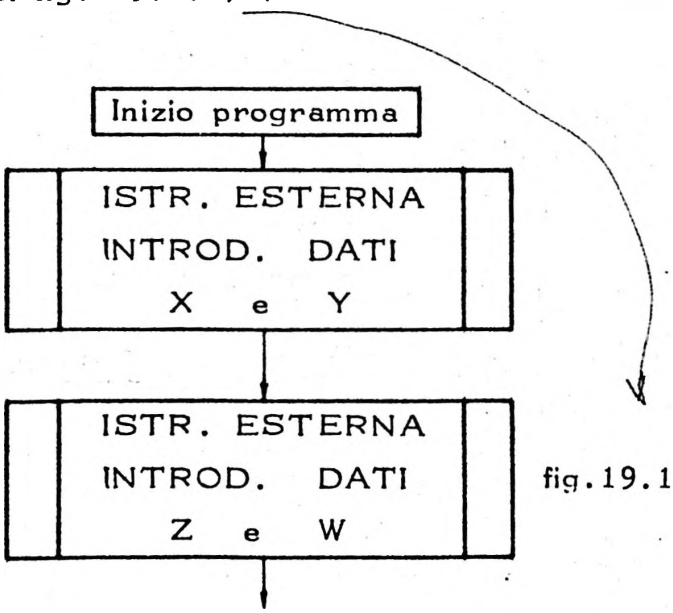


fig.19.1.1

Prima di predisporre le istruzioni che permettono di effettuare le operazioni di somma e sottrazione sui dati introdotti in memoria é necessario fare alcune considerazioni.

Nelle istruzioni di addizione decimale (AD) e di sottrazione decimale (SD) il dato contenuto nel campo E si azzera.

Se si assegna il campo E al dato X nell'addizione e al dato Z nella sottrazione, si deve provvedere affinché questi dati siano utilizzabili per la stampa.

A questo scopo é necessario trasferire sia il dato X che il dato Y, in campi diversi da quelli interessati dalle due operazioni aritmetiche. Ciò é ottenuto con due istruzioni MVC (vedasi fig. 19.1.2).

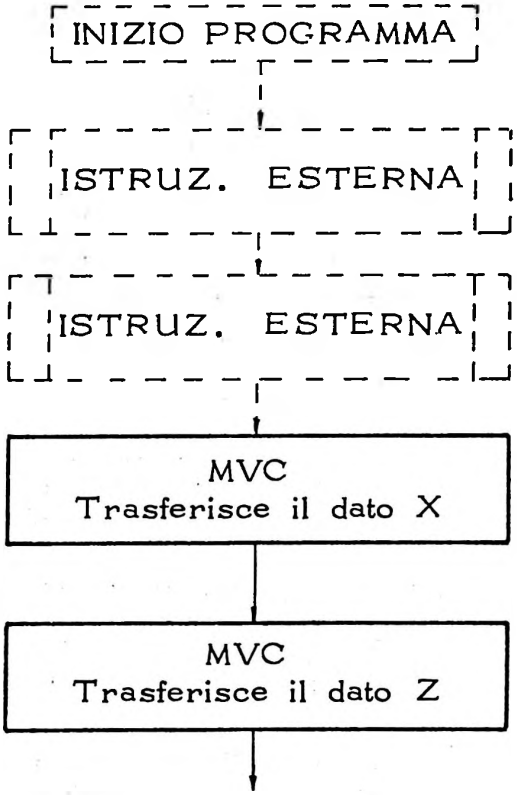


fig. 19.1.2

Ora é possibile fare eseguire le due operazioni aritmetiche :

- la somma con una istruzione AD

- la sottrazione con una istruzione SD (vedasi fig. 19.1.3)

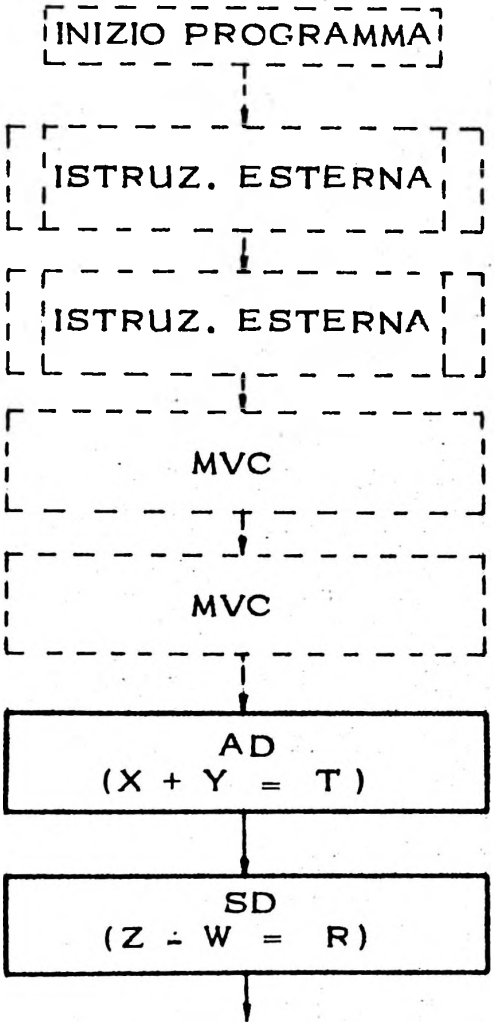


fig. 19.1.3

Con una istruzione esterna che permette di estrarre i dati da memoria e di stamparli, si completa l'elaborazione (vedasi fig. 19.1.4) :

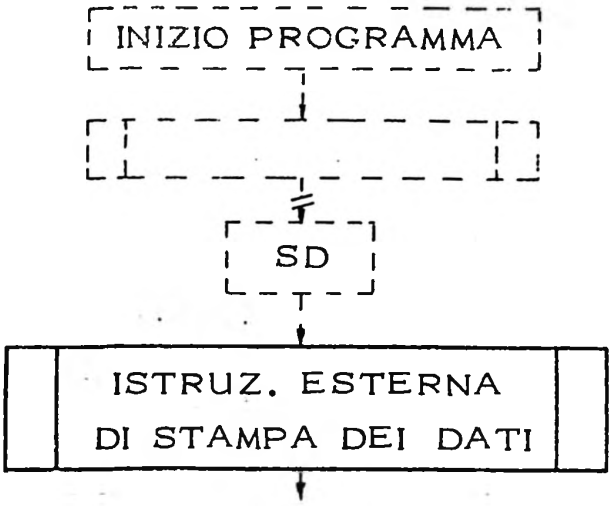


fig. 19.1.4

Il programma così ottenuto permette di introdurre soltanto due schede, una con i dati X e Y, l'altra con i dati Z e W.

Se le schede da elaborare sono molte, si presenta la necessità di iniziare il programma ogni due schede (si supponga che le schede da elaborare siano opportunamente ordinate). Ciò deve avvenire in modo automatico per cui, dopo l'istruzione esterna di stampa, si deve "istruire" l'elaboratore affinché inizi ad interpretare e quindi eseguire, l'istruzione esterna di introduzione dati.

A questo scopo si aggiunge una istruzione JU che fa nuovamente ripetere il programma.

Il flow-chart del programma risulta articolato come in fig. 19.1.5.

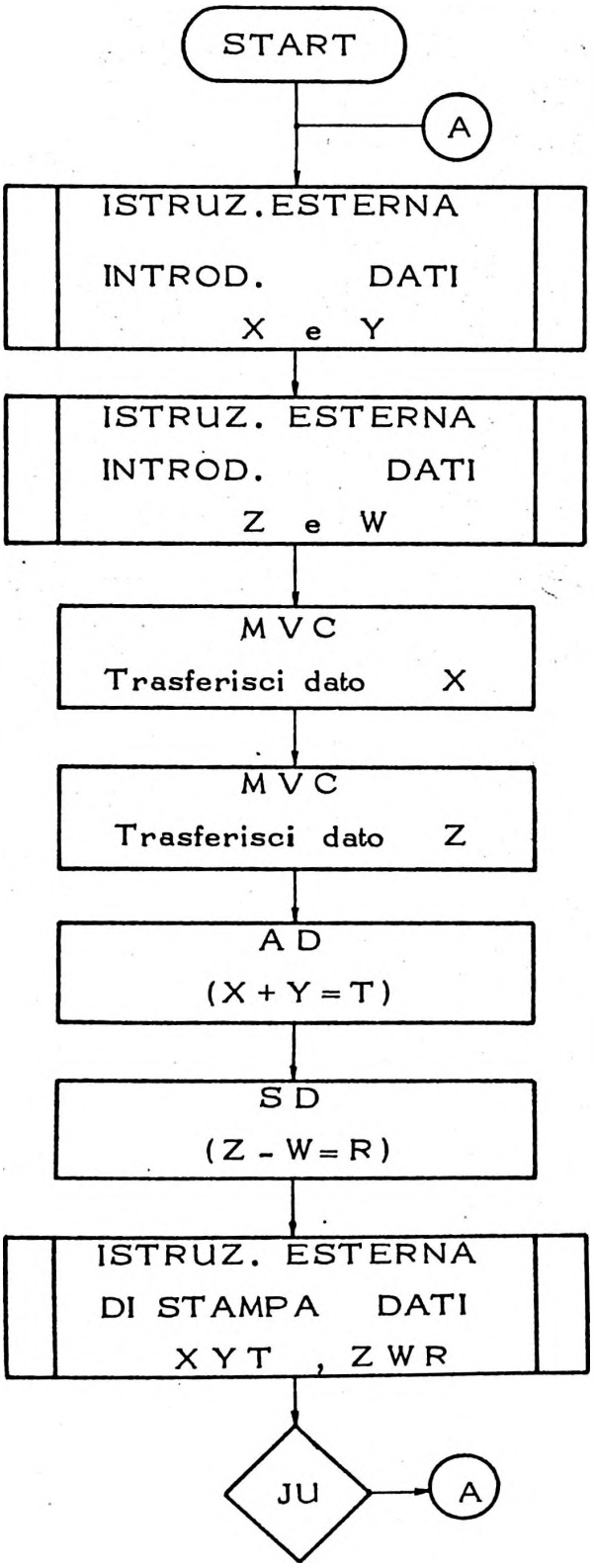

START
A
ISTRUZ.ESTERNA
INTROD. DATI
X e Y
ISTRUZ. ESTERNA
INTROD. DATI
Z e W
M V C
Trasferisci dato X
M V C
Trasferisci dato Z
A D
(X + Y = T)
S D
(Z - W = R)
ISTRUZ. ESTERNA
DI STAMPA DATI
X Y T , Z W R
JU
A

fig. 19.1.5

19.2. REALIZZAZIONE DEL PROGRAMMA

Supposto che :

- ogni istruzione esterna occupi 4 posizioni di memoria,
- la introduzione del programma é fatta a partire dalla posizione 1 di memoria (indirizzo 0/0, 0/0),
- i dati siano introdotti nei seguenti campi (fig. 19.2.1):

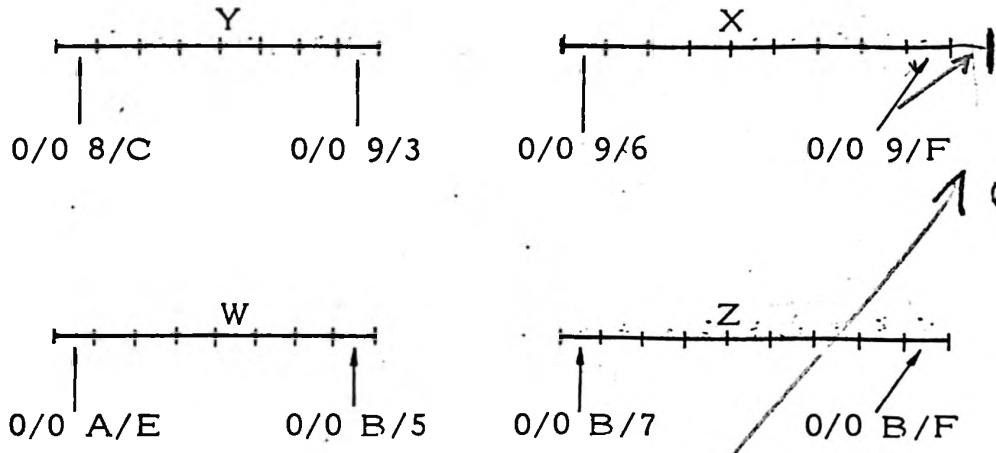


fig. 19.2.1

La MVC che segue queste due istruzioni esterne deve avere il codice di funzione memorizzato nella nona posizione di memoria (indirizzo 0/0 0/8).

DOMANDE

a) - Se si vuole trasferire il dato X nel campo compreso fra gli indirizzi 0/0 8/0 e 0/0 8/9, quali sono i valori dei caratteri che costituiscono tale MVC ?

.....

b) Quali sono i valori dei caratteri che costituiscono la seconda MVC se si vuole trasferire il dato Z a partire dall'indirizzo 0/0 A/3 ?

.....

RISPOSTE

	F	L	IE		IA	
a) - MVC =	D/2	0/9	0/0	8/0	0/0	9/6

	F	L	IE		IA	
b) - MVC =	D/2	0/8	0/0	A/3	0/0	B/7

Dopo aver eseguito le due istruzioni MVC, i dati contenuti nella memoria occupano le posizioni indicate in fig. 19.2.2 :

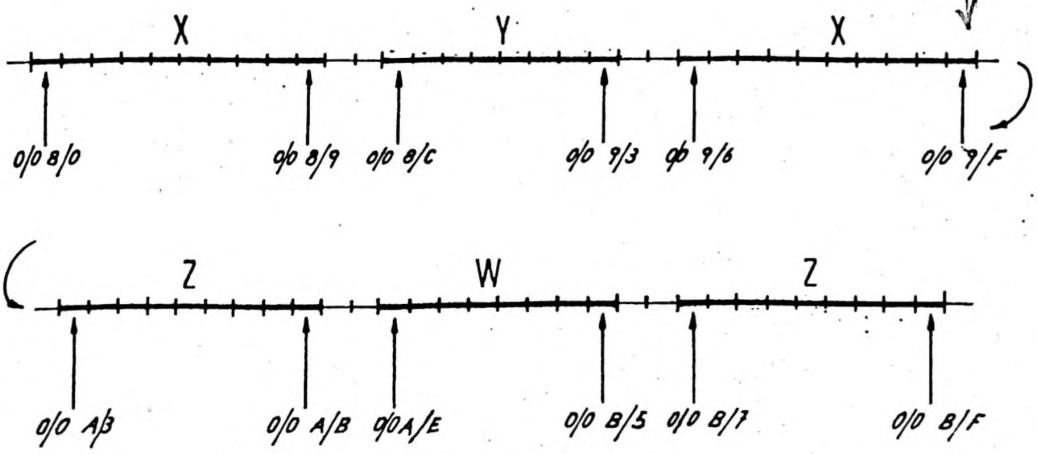


fig. 19.2.2

Con i dati così disposti si possono ora eseguire le operazioni aritmetiche di somma e di sottrazione.

DOMANDE

c) - Come deve essere codificata la AD che esegue la somma X + Y = T ?

F/A 9/7 0/0 9/0 0/0 8/C

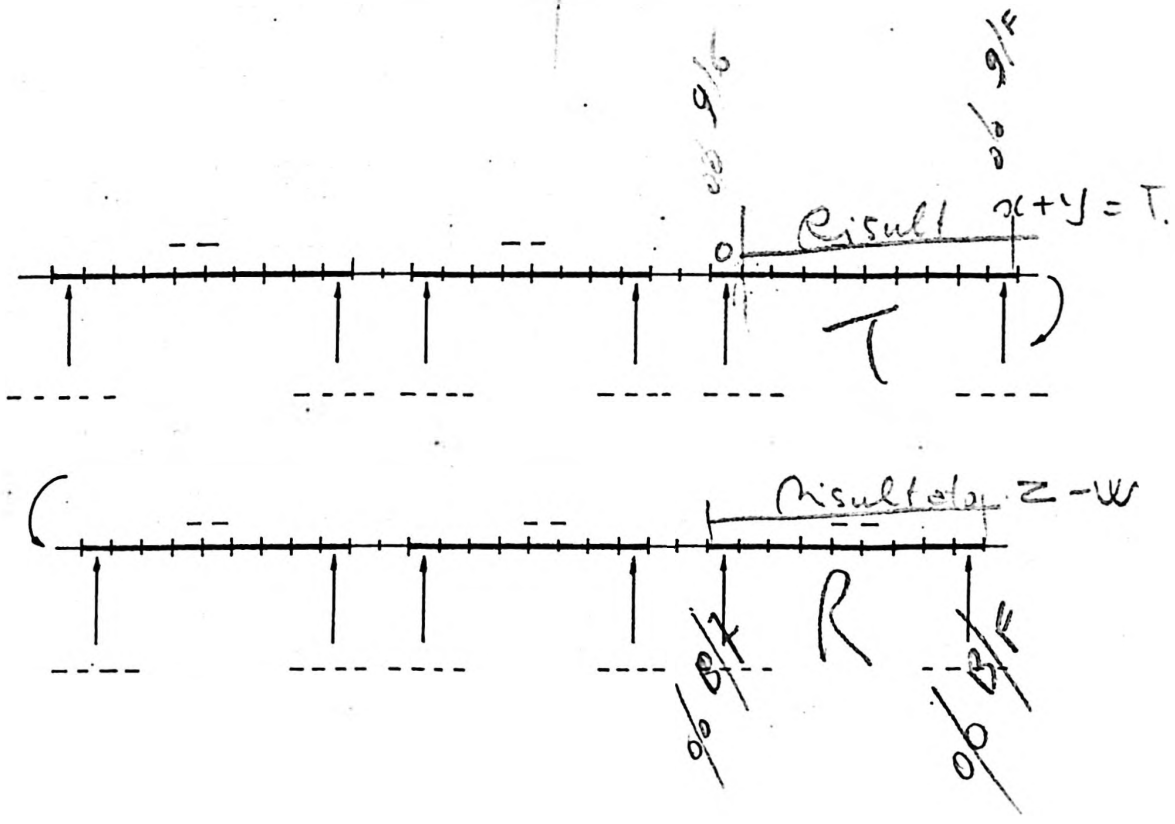
0/0 9/F 0/0 9/3

d) - Quali valori assumono i caratteri componenti la SD che permette di eseguire Z - W = R ?

F/B 8/7 0/0 B/7 0/0 B/3

0/0 B/F 0/0 B/5

e) - Quali posizioni di memoria occupano i risultati ottenuti dalle due operazioni ?

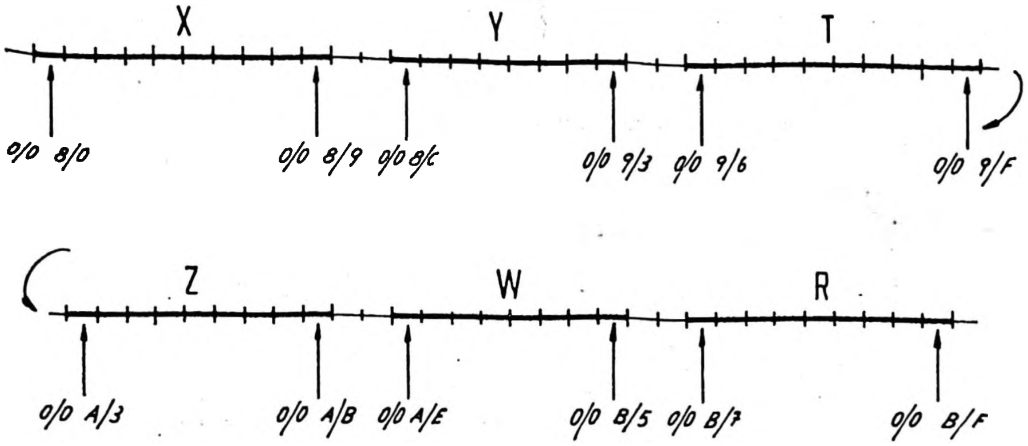


RISPOSTE

	F	LE	LA	IE		IA	
c) -	F/A	9/7		0/0	9/F	0/0	9/3

	F	LE	LA	IE		IA	
d) -	F/B	8/7		0/0	B/F	0/0	B/5

e) -



I dati così ottenuti sono stati elaborati nel modo desiderato e sono preparati per essere stampati. A questo scopo si predispone una istruzione esterna di stampa.

Infine una istruzione di salto incondizionato permette di riprendere il programma dall'inizio.

La istruzione JU e così codificata :

F	F'	IA	
4/7	F/0	0/0	0/0

Ripete le operazioni fino che ci sono schede da leggere e introdurre -

19.3. ANALISI DINAMICA DEL PROGRAMMA

Le istruzioni descritte nel precedente paragrafo costituiscono il seguente programma (fig.19.3.1) :

TIPO di ISTRUZ.	INDIRIZZO CAR. FUNZ. VAL. DEC.	INDIRIZZO CAR. FUNZ. VAL. ESAD.	CARATTERI DELLA ISTRUZIONE F	L	IE	IA		
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI								
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI								
MVC	8	%8	D/2	0/9	%0	8/0	%0	9/6
MVC	14	0/E	D/2	0/8	%0	A/3	%0	B/7
AD	20	1/4	F/A	9/7	%0	9/F	%0	9/3
SD	26	1/A	F/B	8/7	%0	B/F	%0	B/5
ISTRUZIONE ESTERNA DI STAMPA DEI DATI								
JU	36	2/4	4/7	F/0	%0	%0		

fig. 19.3.1

I caratteri del programma sono perforati con schede da secondo un opportuno codice (un carattere occupa due colonne) che ora non si considera perché potrà essere trattato soltanto nella descrizione della istruzione esterna.

In fig. 19.3.2 é riportato lo schema di principio della introduzione del programma di memoria.

Si noti che la memoria rappresentata in figura ha 256 posizioni e ciò soltanto per semplicità di rappresentazione della stessa.

Nelle figure successive i registri indirizzatori sono rappresentati con 12 posizioni mentre i relativi indirizzi sono espressi con due caratteri (16 cifre). Questo perché si suppone che i registri appartengono a un elaboratore dotato di memoria a 4096 posizioni.

RISPOSTE

a) - Alla fine della fase alfa di un'istruzione il registro VI contiene l'indirizzo del carattere di funzione del la istruzione successiva.
Durante la fase beta mantiene inalterato il suo con tenuto.

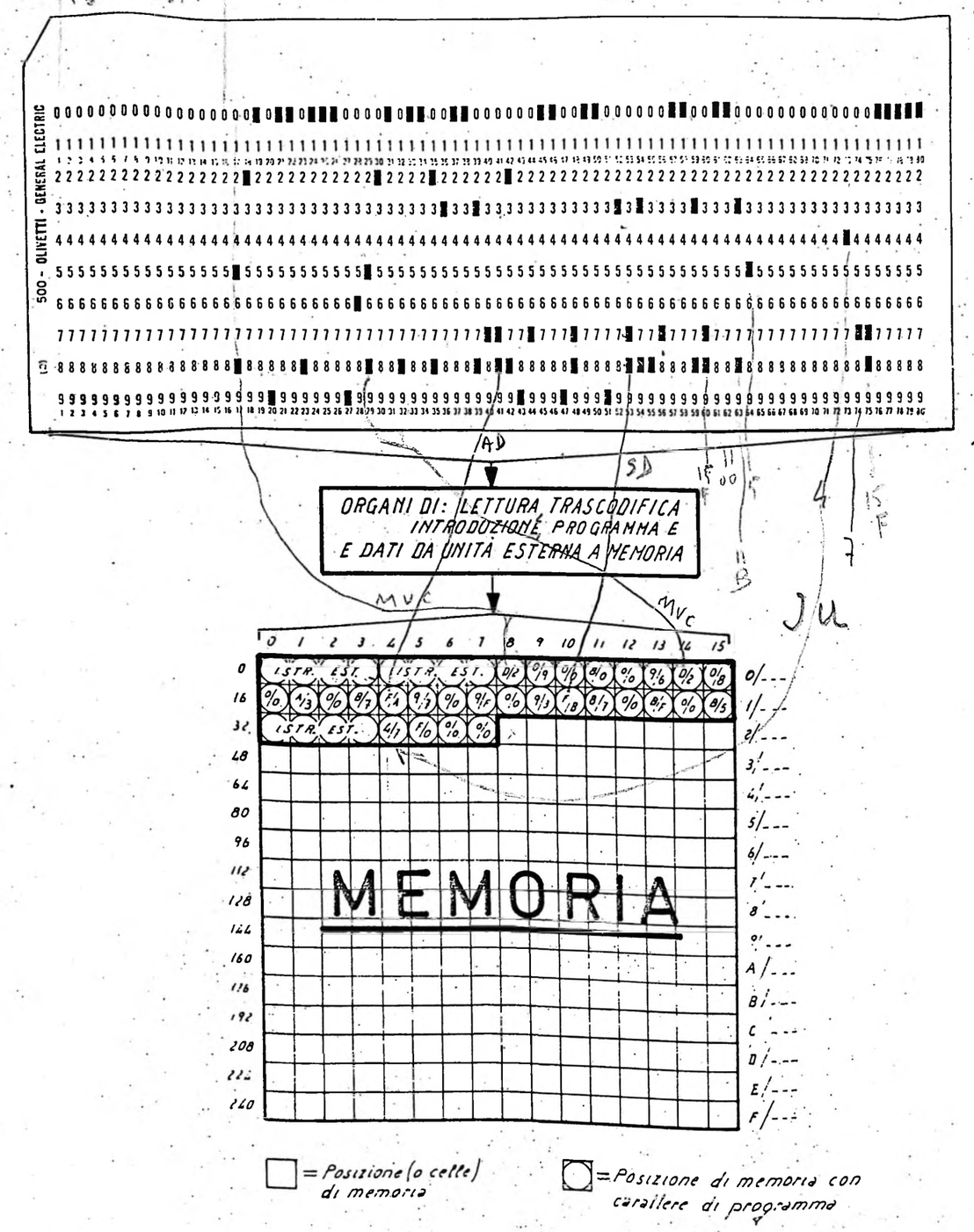
b) - No, perchè le posizioni di memoria comprese fra l'indirizzo 0/0 0/0 e l'indirizzo 0/0 2/7 contengono i caratteri del programma.
Comunque i dati potevano essere introdotti in qual siasi altra zona di memoria.

c) - Perchè nella JU il secondo carattere non esprime la lunghezza dei caratteri da elaborare, ma la condizione per la quale si deve verificare il salto.

d) - VE e VA per indirizzamento dei relativi campi E ed A.
LO per la conta dei caratteri da elaborare RO quale registro di memoria che agisce da tramite nel trasferimento di un carattere da una posizione di memoria ad un'altra.

80 col = 40 costanti

16 col (2)
i primi 8 cer.
istr. est.

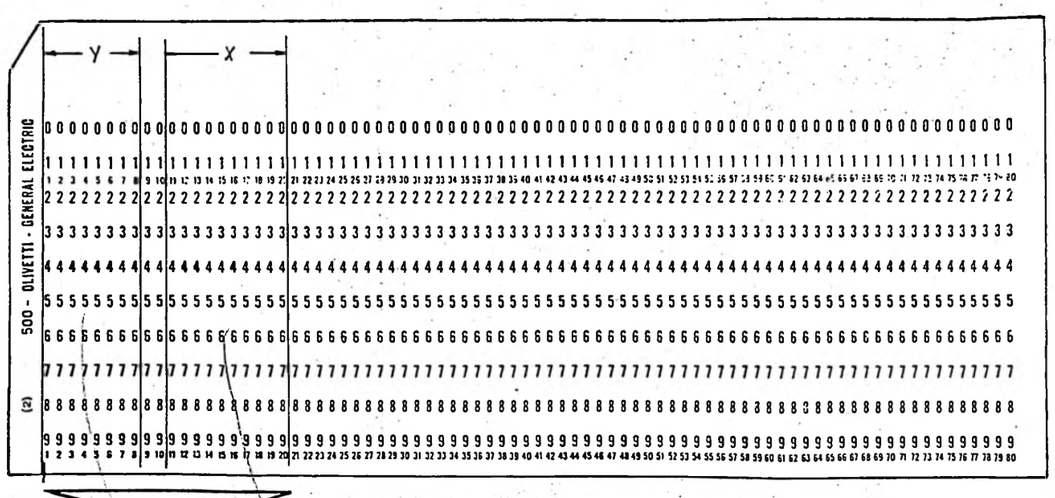


PROGRAMMA

TIPO di ISTRUZ.	INDIRIZZO CAR FUNZ VAL DEC	INDIRIZZO CAR FUNZ VAL ESAD	CARATTERI DELLA ISTRUZIONE F	L	IE	IA
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI						
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI						
MVC	8	0/8	D/2	0/9	0/0	8/0	0/0	9/6
MVC	14	0/E	D/2	0/8	0/0	A/3	0/0	8/7
AD	20	1/4	F/A	9/7	0/0	9/F	0/0	9/3
SD	26	1/A	F/B	8/7	0/0	8/F	0/0	8/5
ISTRUZIONE ESTERNA DI STAMPA DEI DATI						
JU	36	2/4	4/7	F/0	0/0	0/0		

INTRODUZIONE DEL PROGRAMMA
DA LETTORE DI SCHEDE A PARTIRE
DALL'INDIRIZZO 0/0 0/0 DI MEMORIA

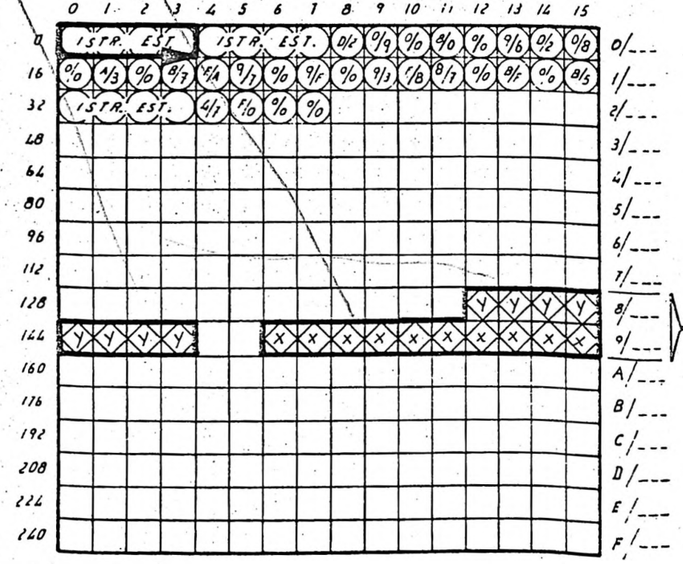
fig. 19.3.2



SCHEDA DATI Y e X

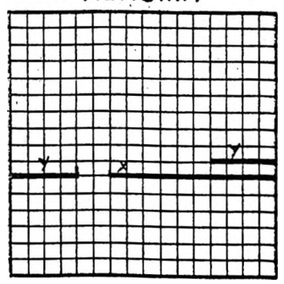
PROGRAMMA

TIPO di ISTRUZ.	INDIRIZZO CAR FUNZ VAL DEC	INDIRIZZO CAR FUNZ VAL ESAD	CARATTERI DELLA ISTRUZIONE F	L	IE	IA
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI						
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI						
MVC	8	0/8	D/2	0/9	0/0	8/0	0/0	9/6
MVC	14	0/E	D/2	0/8	0/0	A/3	0/0	B/7
AD	20	1/4	F/A	9/7	0/0	9/F	0/0	9/3
SD	26	1/A	F/B	8/7	0/0	B/F	0/0	B/3
ISTRUZIONE ESTERNA DI STAMPA DEI DATI						
JU	36	2/4	4/7	F/0	0/0	0/0		



ORGANI DI: LETTURA, TRASCODIFICA E INTRODUZIONE DATI DA UNITÀ ESTERNA A MEMORIA

MEMORIA



□ = Posizione (o celle) di memoria

◯ = Posizione di memoria con carattere di programma

◇ = Posizione di memoria con un carattere di un dato

fig. 19.3.3

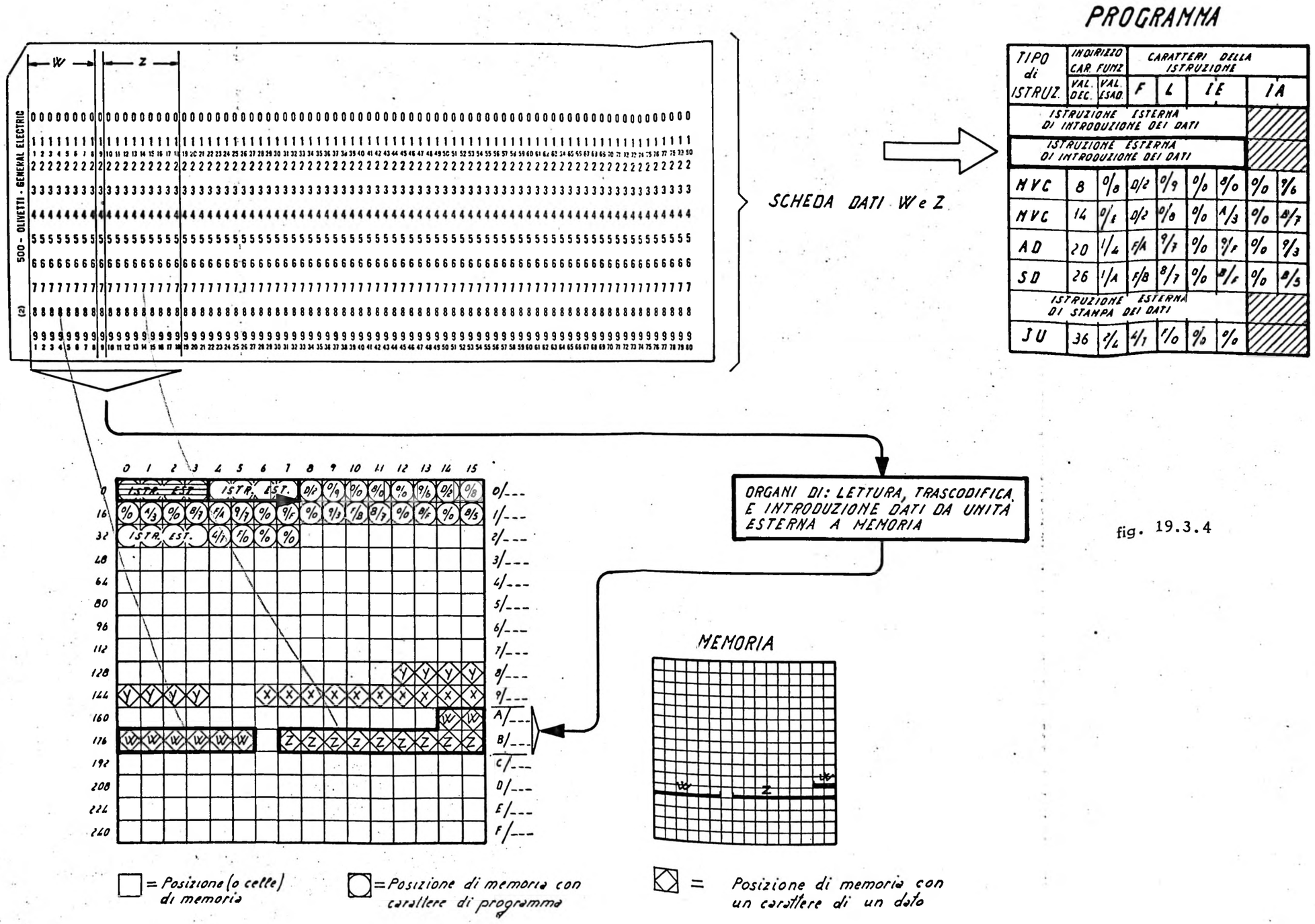

PROGRAMMA
TIPO di ISTRUZ.
INDIRIZZO CAR. FUNZ
VAL. DEC.
VAL. ESAD
CARATTERI DELLA ISTRUZIONE
F
L
IE
IA
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI
MVC 8 0/8 D/2 0/9 0/0 0/0 0/0 9/6
MVC 14 0/E D/2 0/0 0/0 A/3 0/0 B/7
AD 20 1/4 F/A 9/7 0/0 9/F 0/0 9/3
SD 26 1/A F/B B/7 0/0 B/F 0/0 B/5
ISTRUZIONE ESTERNA DI STAMPA DEI DATI
JU 36 2/4 4/1 F/0 0/0 0/0
SCHEDA DATI W e Z
W
Z
ORGANI DI: LETTURA, TRASCODIFICA E INTRODUZIONE DATI DA UNITA ESTERNA A MEMORIA
MEMORIA
= Posizione (o celle) di memoria
= Posizione di memoria con carattere di programma
= Posizione di memoria con un carattere di un dato

fig. 19.3.4

MVC

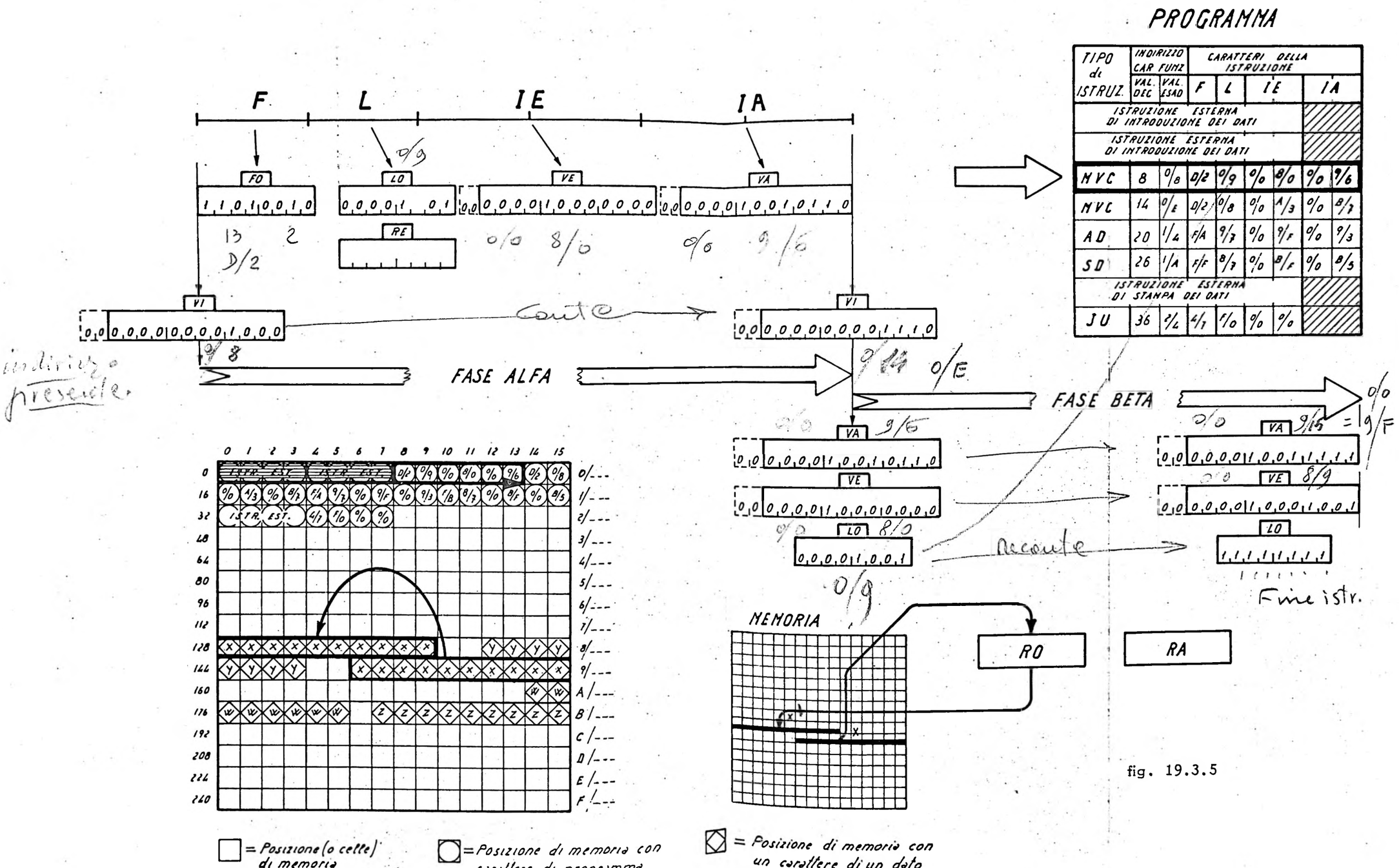
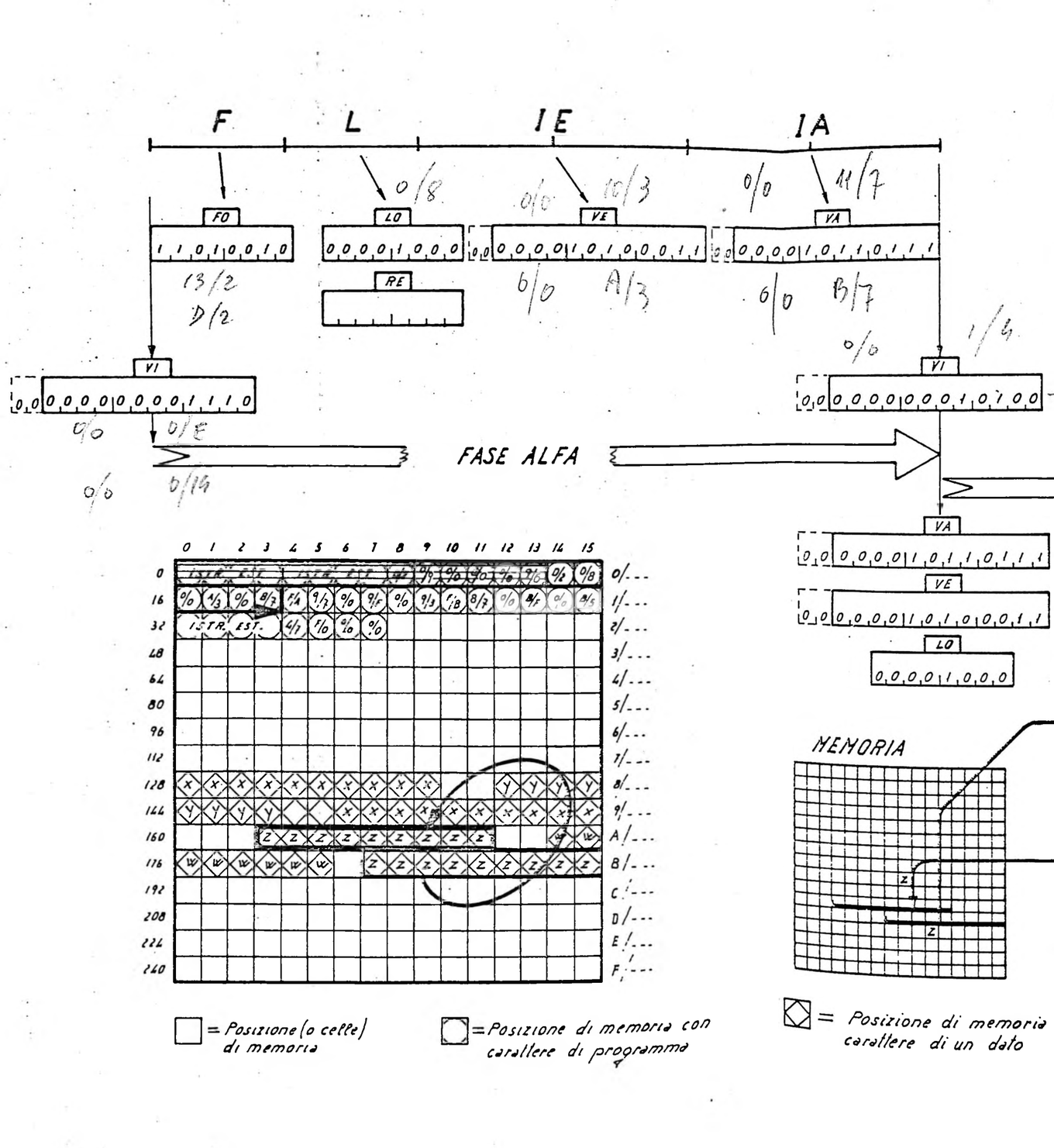

PROGRAMMA
TIPO di ISTRUZ.
INDIRIZZO CAR FUNZ
CARATTERI DELLA ISTRUZIONE
VAL. DEC.
VAL. ESAD.
F
L
IE
IA
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI
MVC 8 0/8 0/2 0/9 0/0 8/0 0/0 9/6
MVC 14 0/E 0/2 0/8 0/0 A/3 0/0 B/7
AD 20 1/4 F/A 9/3 0/0 9/F 0/0 9/3
SD 26 1/A F/F 8/3 0/0 B/F 0/0 B/5
ISTRUZIONE ESTERNA DI STAMPA DEI DATI
JU 36 2/4 4/1 1/0 0/0 0/0
F
L
IE
IA
FO
LO
VE
VA
VI
RE
FASE ALFA
FASE BETA
conta
azzera
indirizzo precedente
Fine istr.
MEMORIA
RO
RA
= Posizione (o celle) di memoria
= Posizione di memoria con carattere di programma
= Posizione di memoria con un carattere di un dato

fig. 19.3.5

PROGRAMMA

TIPO di ISTRUZ.	INDIRIZZO CAR FUNZ		CARATTERI DELLA ISTRUZIONE			
	VAL. DEC.	VAL. ESAD.	F	L	IE	IA
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI						
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI						
NYC	8	0/8	0/2	0/9	0/0	0/0	0/0	9/6
NYC	14	0/E	0/2	0/8	0/0	A/3	0/0	B/7
AD	20	1/4	F/A	9/7	0/0	9/F	0/0	9/3
SD	26	1/A	F/B	8/7	0/0	B/F	0/0	B/5
ISTRUZIONE ESTERNA DI STAMPA DEI DATI						
JU	36	2/4	4/7	1/0	0/0	0/0		

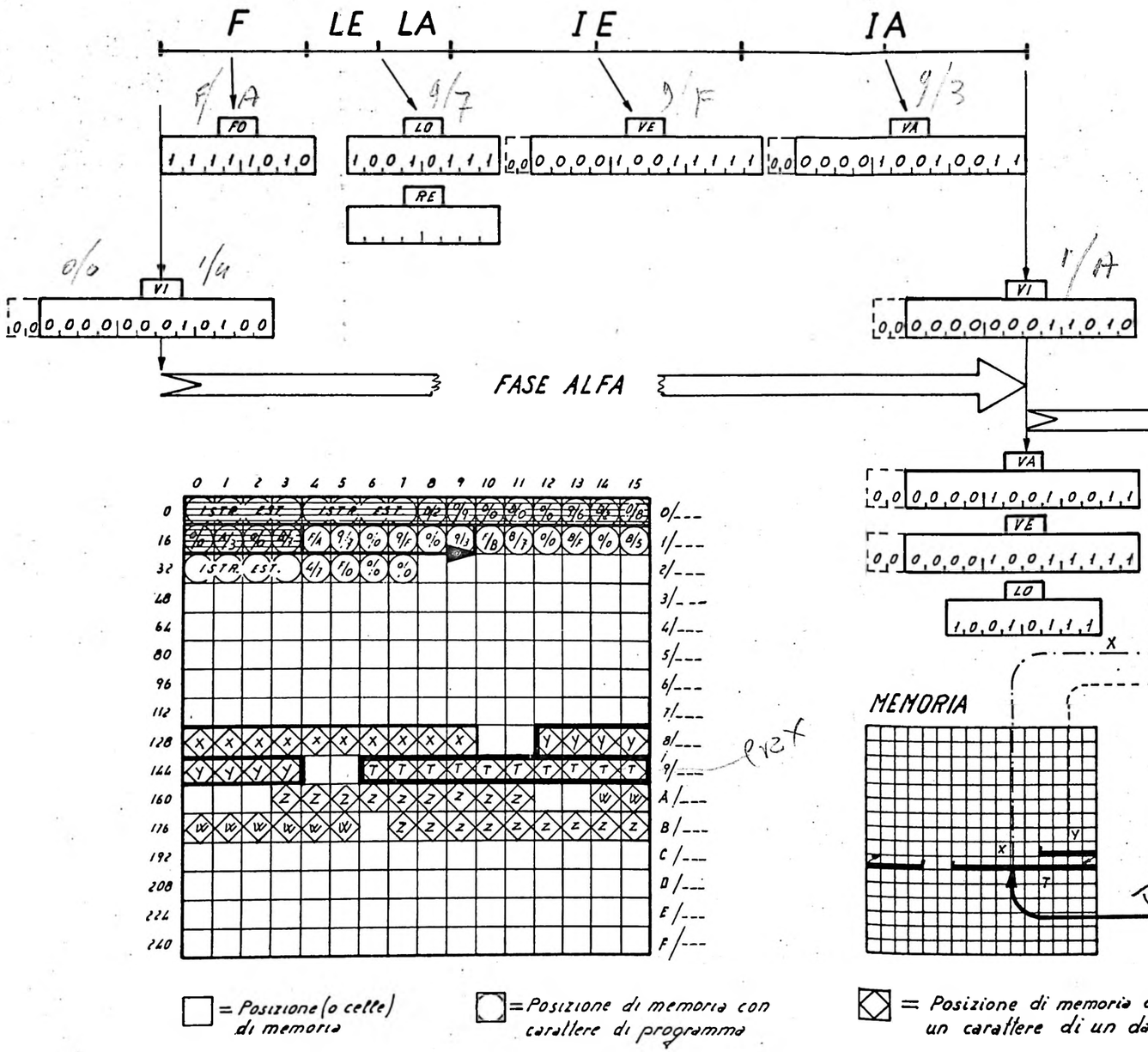


= Posizione (o celle) di memoria

= Posizione di memoria con carattere di programma

= Posizione di memoria con carattere di un dato

fig. 19.3.6



PROGRAMMA

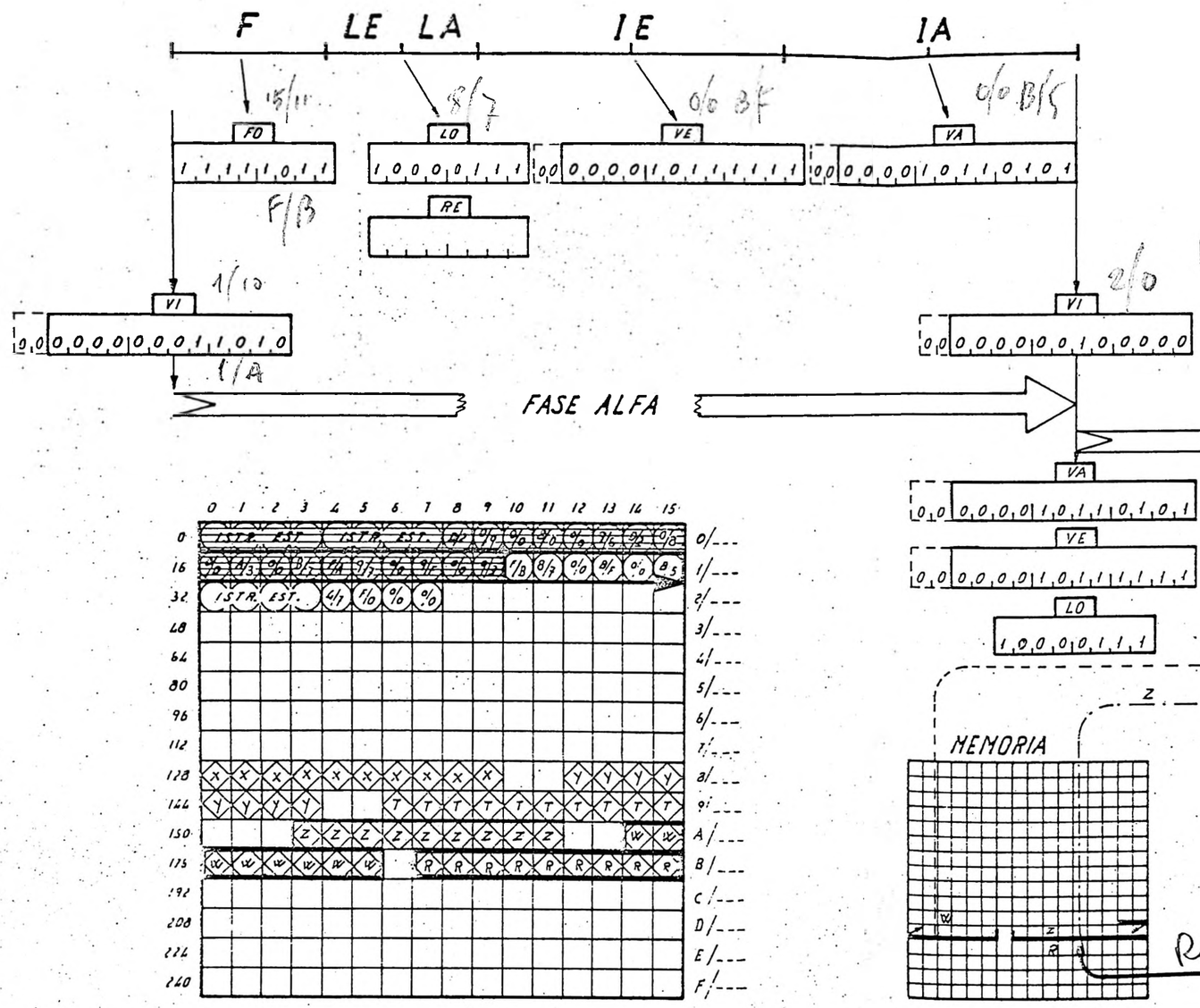
TIPO di ISTRUZ.	INDIRIZZO CAR. FUNZ VAL. DEC.	INDIRIZZO CAR. FUNZ VAL. ESAD.	CARATTERI DELLA ISTRUZIONE F	L	IE		IA	
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI								
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI								
MVC	8	0/8	0/2	0/9	0/0	8/0	0/0	9/6
MVC	14	0/E	0/2	0/8	0/0	A/3	0/0	8/7
AD	20	1/4	F/A	9/7	0/0	9/F	0/0	9/3
SD	26	1/A	F/8	8/7	0/0	B/F	0/0	0/5
ISTRUZIONE ESTERNA DI STAMPA DEI DATI								
JU	36	2/4	4/7	F/0	0/0	0/0		

□ = Posizione (o celle) di memoria

▢ = Posizione di memoria con carattere di programma

⊠ = Posizione di memoria con un carattere di un dato

fig. 19.3.7



PROGRAMMA

TIPO di ISTRUZ.	INDIRIZZO CAR. FUNZ VAL. DEC.	INDIRIZZO CAR. FUNZ VAL. ESAD.	CARATTERI DELLA ISTRUZIONE F	L	IE		IA	
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI								
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI								
MVC	8	0/8	0/2	0/9	0/0	0/0	0/0	9/6
MVC	14	0/E	0/2	0/8	0/0	A/3	0/0	B/7
AD	20	1/4	F/A	9/7	0/0	9/F	0/0	9/3
SD	26	1/A	F/B	8/7	0/0	B/F	0/0	B/5
ISTRUZIONE ESTERNA DI STAMPA DEI DATI								
JU	36	2/4	4/1	F/0	0/0	0/0		

fig. 19.3.8

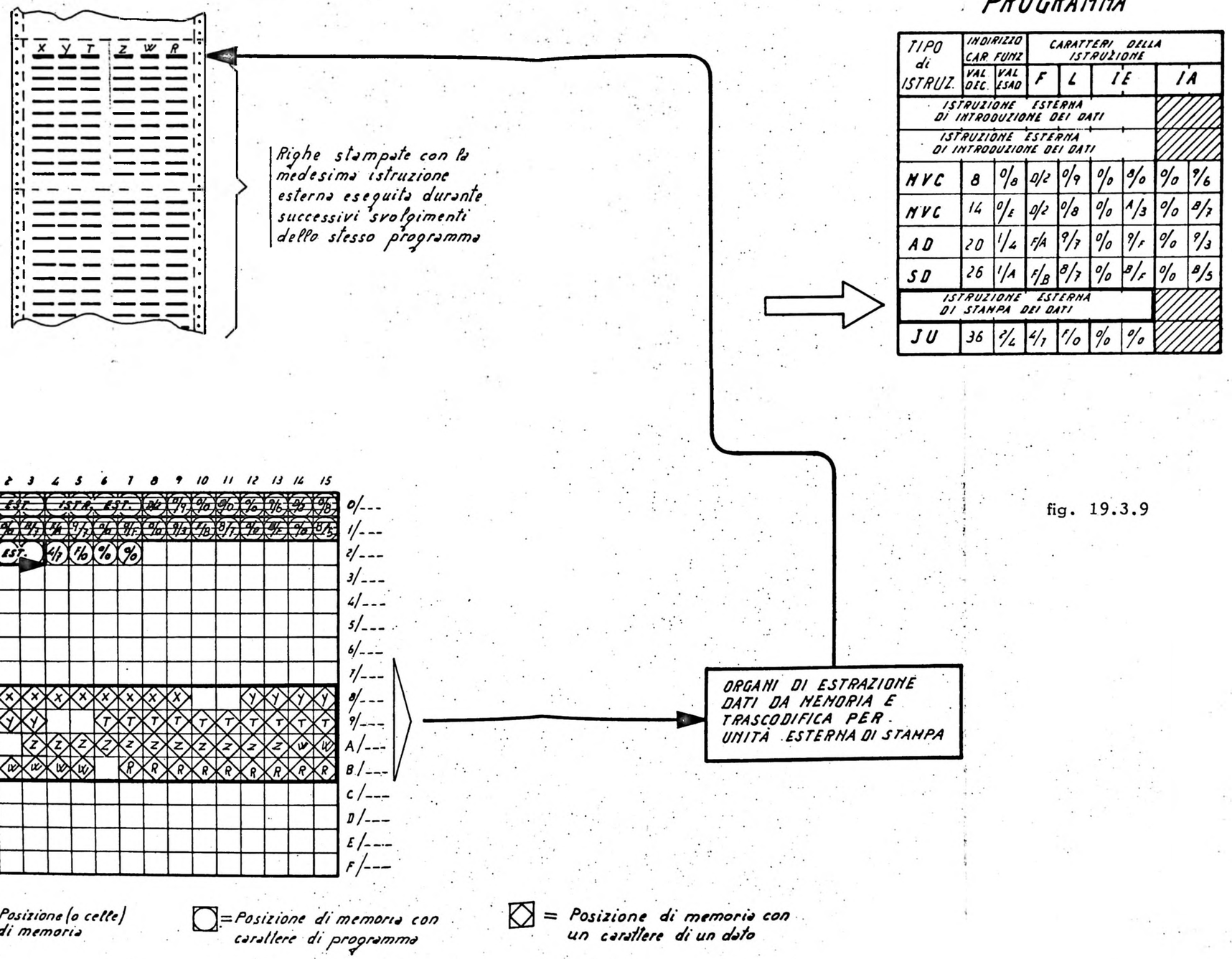
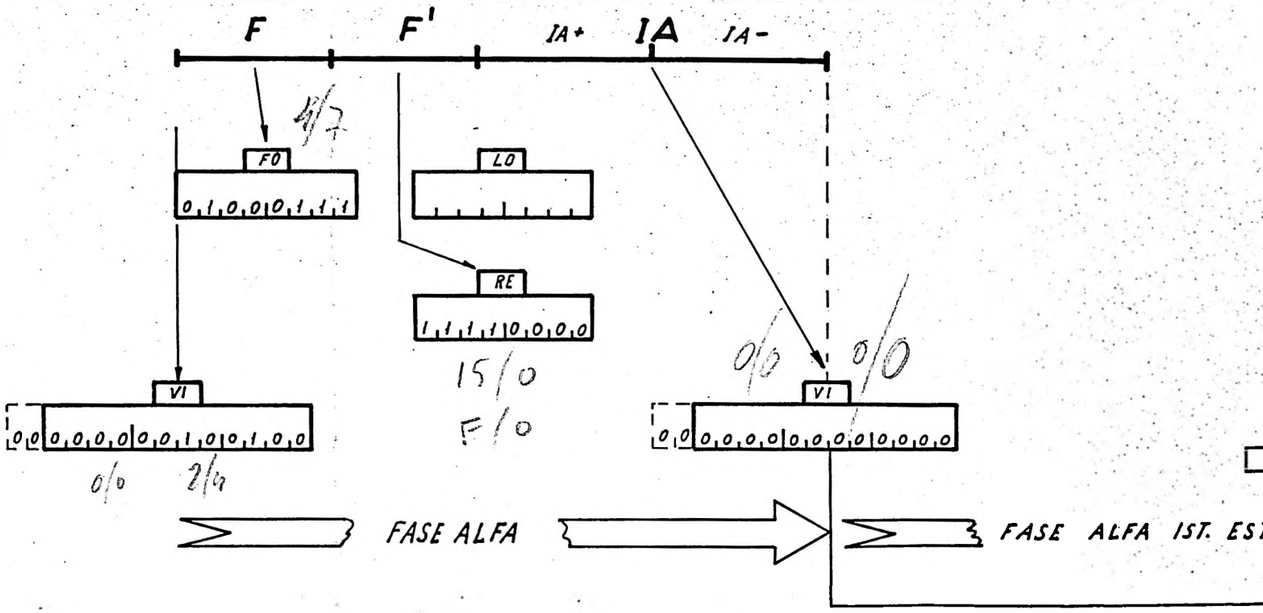

X Y T Z W R
Righe stampate con la medesima istruzione esterna eseguita durante successivi svolgimenti dello stesso programma
PROGRAMMA
TIPO di ISTRUZ.
INDIRIZZO CAR. FUNZ
VAL DEC.
VAL ESAD
CARATTERI DELLA ISTRUZIONE
F
L
IE
IA
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI
MVC 8 0/8 D/2 0/9 0/0 8/0 0/0 9/6
MVC 14 0/E D/2 0/8 0/0 A/3 0/0 B/7
AD 20 1/4 F/A 9/7 0/0 9/F 0/0 9/3
SD 26 1/A F/B B/7 0/0 B/F 0/0 B/5
ISTRUZIONE ESTERNA DI STAMPA DEI DATI
JU 36 2/4 4/7 F/0 0/0 0/0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0/---
1/---
2/---
3/---
4/---
5/---
6/---
7/---
8/---
9/---
A/---
B/---
C/---
D/---
E/---
F/---
ORGANI DI ESTRAZIONE DATI DA MEMORIA E TRASCODIFICA PER UNITÀ ESTERNA DI STAMPA
= Posizione (o celle) di memoria
= Posizione di memoria con carattere di programma
= Posizione di memoria con un carattere di un dato

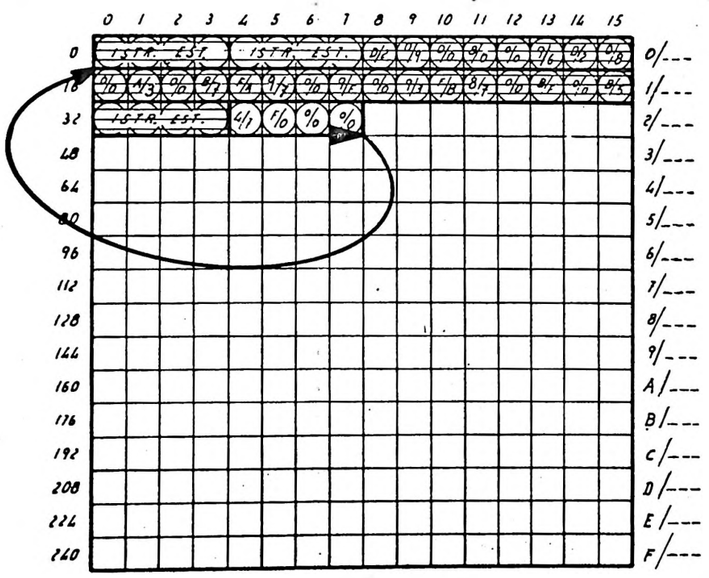
fig. 19.3.9



PROGRAMMA

TIPO di ISTRUZ.	INDIRIZZO CAR. FUNZ		CARATTERI DELLA ISTRUZIONE					
	VAL. DEC.	VAL. ESAD.	F	L	IE		IA	
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI								
ISTRUZIONE ESTERNA DI INTRODUZIONE DEI DATI								
MVC	8	0/8	D/2	0/9	0/0	B/0	0/0	9/6
MVC	14	0/E	D/2	0/8	0/0	A/3	0/0	B/7
AD	20	1/4	F/A	9/7	0/0	9/F	0/0	9/3
SD	26	1/A	F/B	B/7	0/0	B/F	0/0	B/5
ISTRUZIONE ESTERNA DI STAMPA DEI DATI								
JU	36	2/4	4/7	F/0	0/0	0/0		

CARICAMENTO DI **VI** DURANTE LA FASE ALFA ISTRUZIONE **JU** (Salto incondizionato) AVVIENE SUBITO DOPO LA LETTURA DI **IA**



□ = Posizione (o celle) di memoria

◯ = Posizione di memoria con carattere di programma

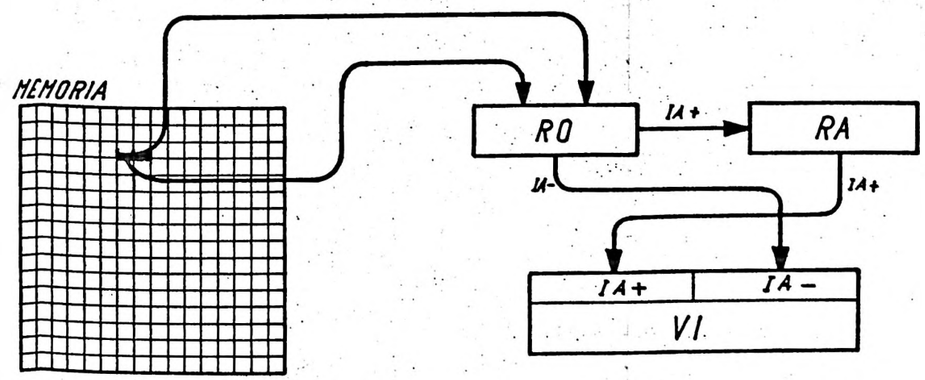


fig. 19.3.10

LEZIONE 20

Tempo di studio
30 minuti

20. INTERPRETAZIONE DI PROGRAMMI

20.1. In questa lezione si esaminano alcuni programmi elementari per meglio comprendere l'aspetto dinamico delle istruzioni del GE-115.

I programmi sono riportati in un apposito tracciato denominato foglio programma, per rendere più semplice l'interpretazione del programma in esame.

DOMANDE

Si descriva brevemente ognuno dei cinque programmi :

a) - Programma ① Salto all'indirizzo o/o o/o 1°pos Mem
.....
.....

b) - Programma ② Trasferisce il contenuto 4/7
all'indirizzo IE o/o o/o che è la posizione
nel caso F ... e questa istruzione
inserita in sede JU-4/7 F/o e sede
del IE o/o o/o

c) - Programma ③ JU salta a MVI il quale fa
all'indirizzo 0004 il codice F → O/A (HLT)
quindi nella istruzione seguente salta all'indirizzo 0004
e si arresta il programma.

d) - Programma ④
MVI trasferisce 5/9 in L nella linea seguente
CMI confronta con se stesso se = salta a HLT.
Se ≠ va a (JNE) riporta a MVI ecc. (JE)

e) - Programma ⑤

MVC trasferisce 2 caratteri dalla IE 0/9 0/0 a 0/4 0/1
prelevati da IA 0/3 F/6 (corrisponde HLT. e 0/4 0/0

Le 1:— MV1 trasferisce 0/7 all'Ind. 0/3 F6 così
questa istruzione diventa NOP2.

Le 2: MVI. porta in 0/3 F/7. i codici 00 per cui
compirà la NOP2 (che prima era un HLT
poiché .NOP2.... arriva la CMI che compirà
e controllerà che all'indirizzo 0/4 0/0 ci sia
stato trasferito O/A. , se ciò è avvenuto
la JNE non si effettua e così il programma
si ferma all'HLT.
qualora il confronto non è = la JNE rimanda
e rifarà daccapo il programma

indirizzo progressivo
del codice di ripetizione
di quella istruzione
o linea di ripetizione

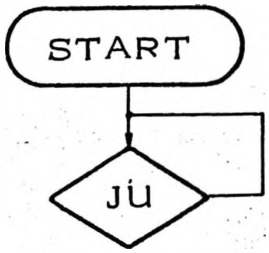
FOGLIO PROGRAMMA GE-115

F	L/F	IE+	IE−	IA+	IA−		Cod. simb.	Indir. esadec.		Ind. dec.	N o t e
4 7	F 0	0 0	0 0				JU	0 0	0 0	0000	①
9 2	4 7	0 0	0 4				MVI	0 0	0 0	0000	②
5 3	F 0	0 0	0 0			JU	xxxx	0 0	0 4	0004	
4 7	F 0	0 0	0 6				JU	0 0	0 0	0000	③
5 3	0 0					HLT.	xxxx	0 0	0 4	0004	
9 2	0 A	0 0	0 4				MVI	0 0	0 6	0006	
4 7	F 0	0 0	0 4				JU	0 0	0 A	0010	
9 2	5 9	0 3	E D				MVI	0 3	E 8	1000	④
9 5	4 7	0 3	E D				CMI	0 3	E C	1004	
4 7	2 0	0 3	F 8				JE	0 3	F 0	1008	
4 7	D 0	0 3	E 8				JNE	0 3	F 4	1012	
0 A	0 0						HLT	0 3	F 8	1016	
	0 = Real.	0/4 - 0/4	0/0 0/1								
D 2	0 1	0 4	0 0	0 3	F 6		MVC	0 3	E 8	1000	⑤
9 2	0 7	0 3	F 6				MVI	0 3	E E	1006	
9 2	0 0	0 3	F 7				MVI	0 3	F 2	1010	
0 A	0 0					NOP2	HLT	0 3	F 6	1014	
9 5	0 A	0 4	0 0				CMI	0 3	F 8	1016	
4 7	D 0	0 3	E 8				JNE	0 3	F C	1020	
0 7	0 0						NOP2	0 4	0 0	1024	
0 A	0 0					HLT					

RISPOSTE

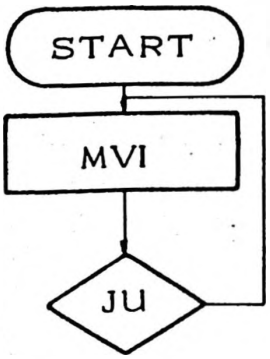
a) - Programma (1)

L'istruzione JU posta all'indirizzo 0/0 0/0 di memoria cicla sempre su se stessa, poiché il codice F' F/0 esprime un salto incondizionato all'indirizzo, espresso da IE^+ ed IE^- (0/0 0/0),



b) - Programma (2)

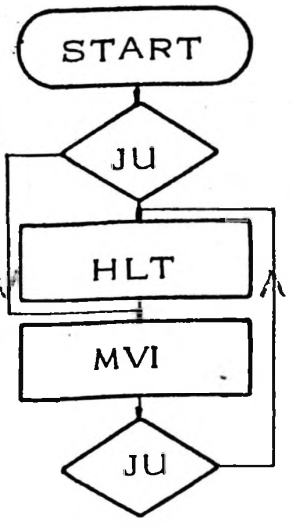
La MVI pone all'indirizzo del carattere di funzione dell'istruzione seguente il codice 4/7 corrispondente ad una JU. Ciò consente, se il trasferimento é esatto, di far riciclare la sequenza MVI - JU in quanto quest'ultima rimanda alla MVI incondizionatamente (F' = F/0, IE^+ IE^- = 0/0, 0/0)



c) - Programma (3)

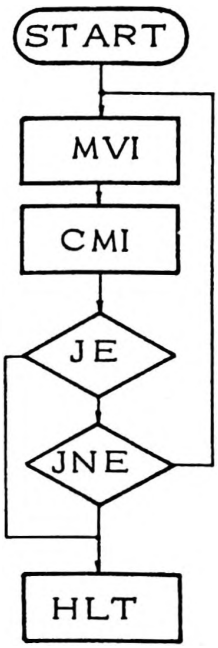
La JU salta l'istruzione da definire facendo eseguire la MVI. Questo trasferimento completa la seconda istruzione con il carattere 0/A (codice di funzione dell'istruzione HLT) in posizione 0/0 - 0/4.

La seconda JU va a fermare l'elaborazione ritornando alla istruzione HLT ed eseguendola



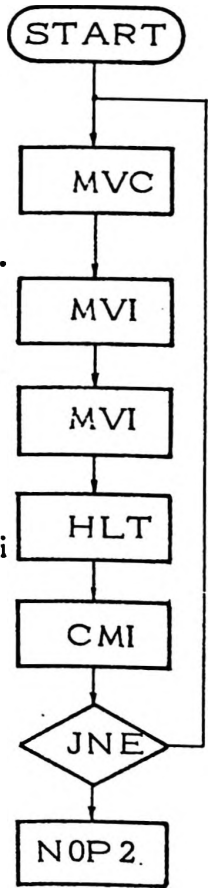
d) - Programma ④

La MVI compila la CMI nella posizione del carattere di confronto ponendo 5/9.
Il confronto viene eseguito dal carattere 5/9 con se stesso.
Se si ha segnalazione di eguaglianza la JE (codice F'= 2/0) ci porta all'ALT.
Se per errore di trasferimento o di confronto cioò non avvenisse l'istruzione JNE (codice F'= D/0) ci riporta a rifare la TFC.
Questo si verifica se la JE non trova la condizione verificata (confronto ≠).



e) - Programma ⑤

La MVC trasferisce l'istruzione HLTposta a 0/3 - F/6 all'indirizzo 0/4 - 0/0.
Con ciò si sostituisce la N0P2.
Le due MVI sostituiscono all'indirizzo 0/3 - F/6 l'istruzione HLT con una N0P2.
La CMI fa il confronto tra il suo carattere (0/A) e quello posto all'indirizzo 0/4 - 0/0 che deve corrispondere al codice di funzione dell'istruzione HLT.
Se il confronto dà eguaglianza significa che la MVC ha fatto il trasferimento esatto, e arriva all'HLT.
Se invece non si ottiene risultato uguale si ripete il trasferimento tramite la JNE.



LEZIONE 21

Tempo di studio
ore 1 e 30 minuti

21. ESAME DI UN PROGRAMMA

21.1. In questa lezione, conclusiva dei principi di programmazione già esposti, si esamina un programma che in particolare utilizza alcuni pulsanti ed alcune indicazioni luminose di console operativa.

Si hanno due dati : Y e X, con Y sempre maggiore di X. Si vuole eseguire l'operazione Y - X = Z e quindi, a seconda del posizionamento dei pulsanti SWITCH 1 (SW1) e SWITCH 2 (SW2), si vogliono eseguire diverse operazioni sul dato Z..

Tali operazioni sono, a seconda se i pulsanti sono posizionati (1) e non posizionati (0), le seguenti : .

Con SW1 = 1 e SW2 = 0	si calcola il 2 ‰ di Z ·
Con SW1 = 0 e SW2 = 1	si calcola il 4 ‰ di Z .
Con SW1 = 0 e SW2 = 0	non si calcola nulla ·
Con SW1 = 1 e SW2 = 1	non esegue nessuna operazione. L'elaborazione si arresta con l'indicazione luminosa OPERATOR CALL accesa per indicare tale avvenimento.

Si vuole anche stampare il valore di Z e l'eventuale parte millesimale.

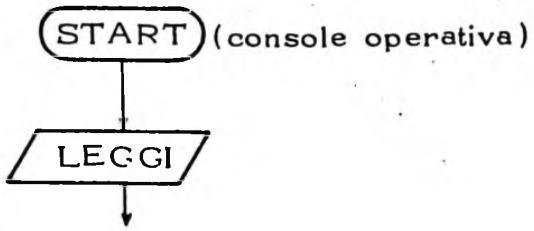
Nello studio di un programma bisogna tenere presente che lo stesso può essere modificato ed ampliato per nuove esigenze di elaborazione.

A questo scopo si inseriscono nel programma delle istruzioni NOP 2 che servono a mantenere libere alcune posizioni di memoria le quali potranno essere successivamente occupate da altre operazioni che determinano le modifiche del programma.

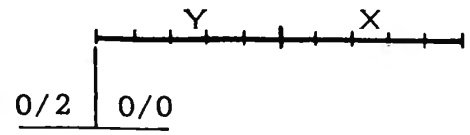
Un esempio di modifica di un programma é illustrato alla fine di questa lezione.

21.2. Si inizia lo studio analizzando il flow-chart del programma..

Si inizia con un'istruzione esterna di introduzione dei dati da elaborare. Questa istruzione esterna, che si denomina brevemente "LEGGI" occupa quattro posizioni di memoria.



Si supponga che i dati siano introdotti uno adiacente all'altro in memoria ed occupino le seguenti posizioni:

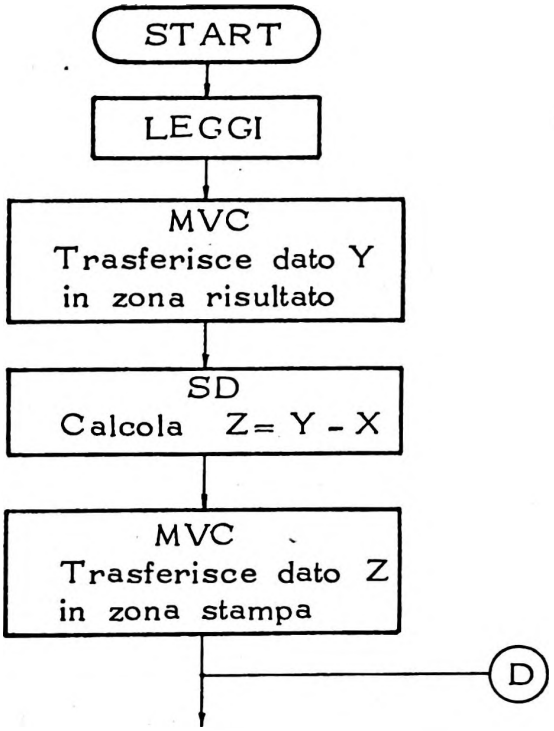


Si voglia inoltre mantenere inalterate le due zone di memoria contenenti i dati stessi; sino alla fine del programma.

Si rende allora necessario trasferire il dato Y (campo E) in un'altra zona di memoria prima di eseguire l'operazione Y - X.

A questo scopo si esegue un'istruzione MVC che trasferisce il dato Y in una zona di memoria che si denomina zona risultato.

Dopo tale operazione si può eseguire la sottrazione fra il dato Y in zona risultato (campo E) ed il dato X (campo A) e trasferirne il risultato in una zona di memoria riservato ai dati da stampare.



A questo punto si deve esaminare le posizioni assunte dai due pulsanti SW1, SW2 con le istruzioni JS1 e JS2.

Il flow-chart complessivo dei quattro diversi casi che si possono presentare a seconda del posizionamento dei pulsanti SW1 e SW2 si riporta in Fig. 21.2.1.

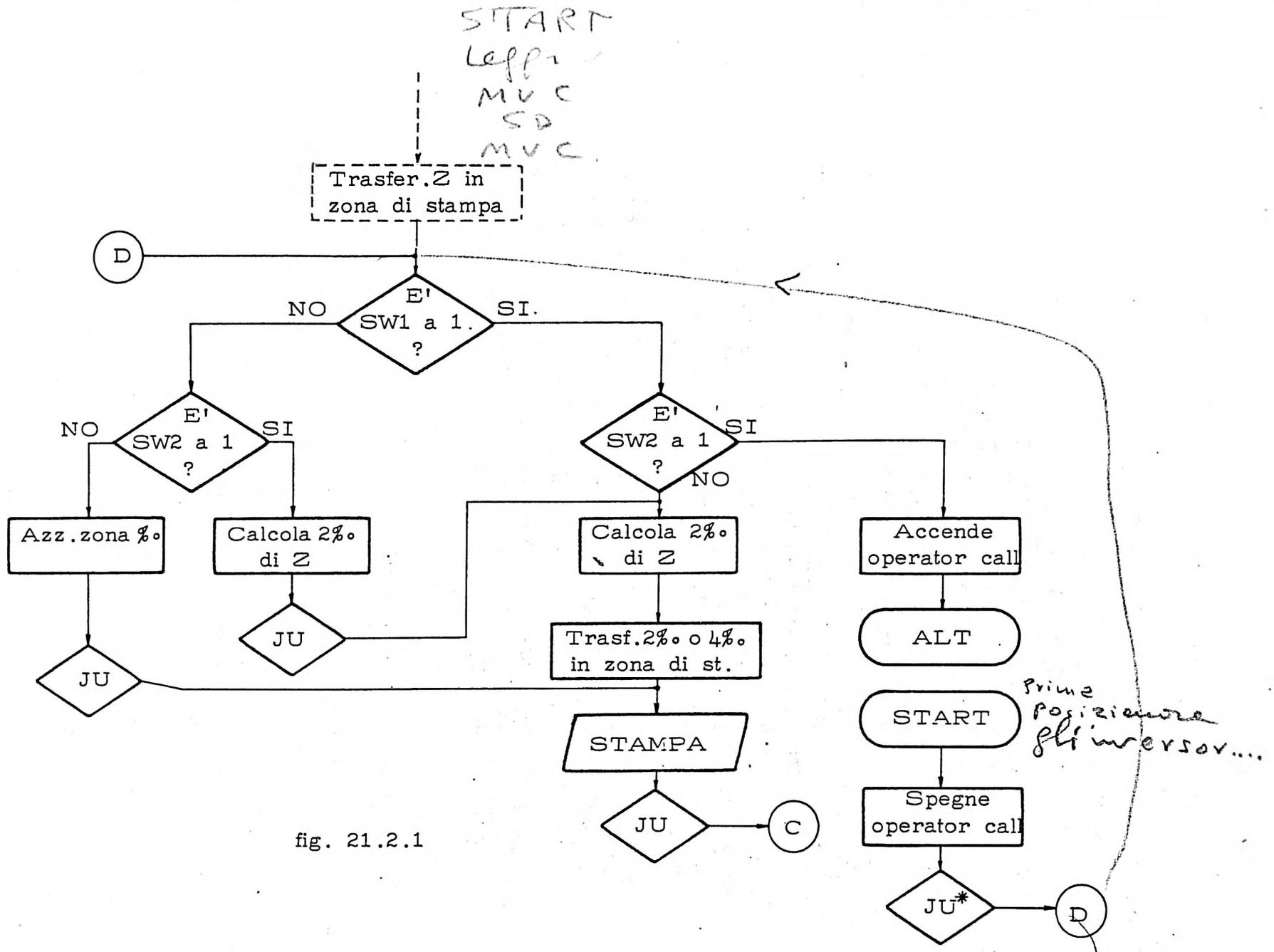
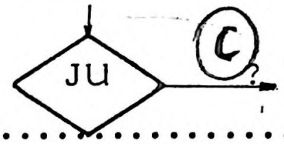

START
Leggi
MVC
SD
MVC.
Trasfer.Z in zona di stampa
D
E' SW1 a 1. ?
NO
SI.
E' SW2 a 1 ?
NO
SI
Azz.zona %o
Calcola 2%o di Z
JU
JU
E' SW2 a 1 ?
SI
NO
Calcola 2%o di Z
Trasf.2%o o 4%o in zona di st.
Accende operator call
ALT
START
Prima posizionare gli inversor...
Spegne operator call
JU*
D
STAMPA
JU
C

fig. 21.2.1

DOMANDE

a) - Dove deve saltare il programma dopo l'istruzione JU indicata con :



Prossima istr. esterna ?
Ai Leggi .

.....
.....

b) - Quale operazione é necessario eseguire dopo l'arresto del programma con l'istruzione HLT, per poter ottenere uno dei risultati previsti dal programma ?

Posizionare gli jumpers START su Reset
interruttori .....

c) - Quale scopo ha l'istruzione JU che segue l'istruzione "spegne operator call" ?

.....
.....

Al segnale di operator che
gli interruttori non sono posizionati
bene . quindi si posiz. bene e si
riporta con istr. nel punto D .

RISPOSTE

a) - Ad inizio programma. col reset

b) - Si deve posizionare in modo corretto gli SW1 e SW2 e cioè in una diversa posizione da :

SW1 = 1 SW2 = 1

c) - La JU ha lo scopo di esplorare nuovamente il posizionamento dei pulsanti per verificare se sono stati predisposti opportunamente dopo che il programma ha segnalato che il percorso eseguito non conduce a nessun risultato.
Il flow-chart completo delle istruzioni sinora esaminate e riportato in Fig. 21.2.2.

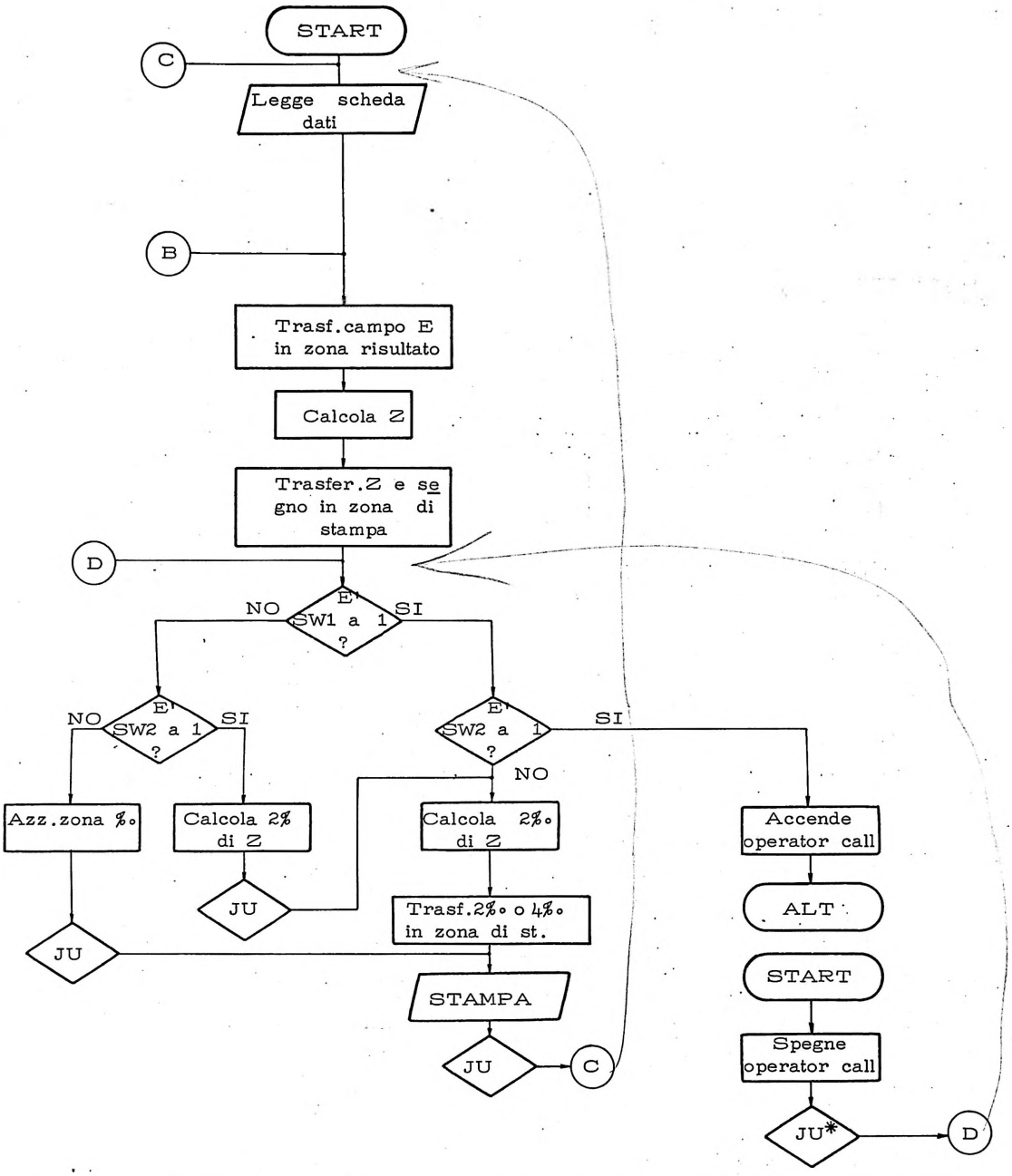

START
C
Legge scheda dati
B
Trasf.campo E in zona risultato
Calcola Z
Trasfer.Z e segno in zona di stampa
D
E' SW1 a 1 ?
NO
SI
E' SW2 a 1 ?
NO
SI
Azz.zona %o
Calcola 2% di Z
JU
JU
E' SW2 a 1 ?
SI
NO
Calcola 2%o di Z
Trasf.2%o o 4%o in zona di st.
STAMPA
JU
C
Accende operator call
ALT
START
Spegne operator call
JU*
D

fig. 21.2.2

21.3. Nella premessa fatta nel paragrafo 21.1 si era supposto che il dato Y fosse sempre maggiore del dato X. Se questa ipotesi non sussiste Y può essere maggiore, eguale o minore a X.

Ricordando che nell'istruzione di sottrazione decimale (SD) non é prevista la correzione del risultato se il minuendo é minore del sottraendo, si deve predisporre l'eventuale correzione per mezzo del programma.

A questo scopo, prima di eseguire il calcolo di Z, ed esattamente subito dopo l'introduzione dei dati in memoria é necessario esaminare, con un confronto, quale dei due dati X e Y é maggiore. Quindi si deve assegnare il campo E al dato maggiore in modo da ottenere un risultato corretto.

In questo modo non risulterebbe evidente se il risultato si riferisce a Y - X per Y > X, oppure X - Y per X < Y.

Si ovvia a questo inconveniente indicando ad esempio, il risultato con un segno + a destra del risultato se questo si riferisce a Y - X, con un segno - per X - Y.

A questo scopo dopo la "leggi" si introduce una JRT che fa uscire dal programma principale e va ad eseguire un sottoprogramma detto anche "sub-routine".

Il flow-chart che riassume questo insieme di istruzioni che si denomina "sub-routine segno" é rappresentato in fig. 21.3.1.

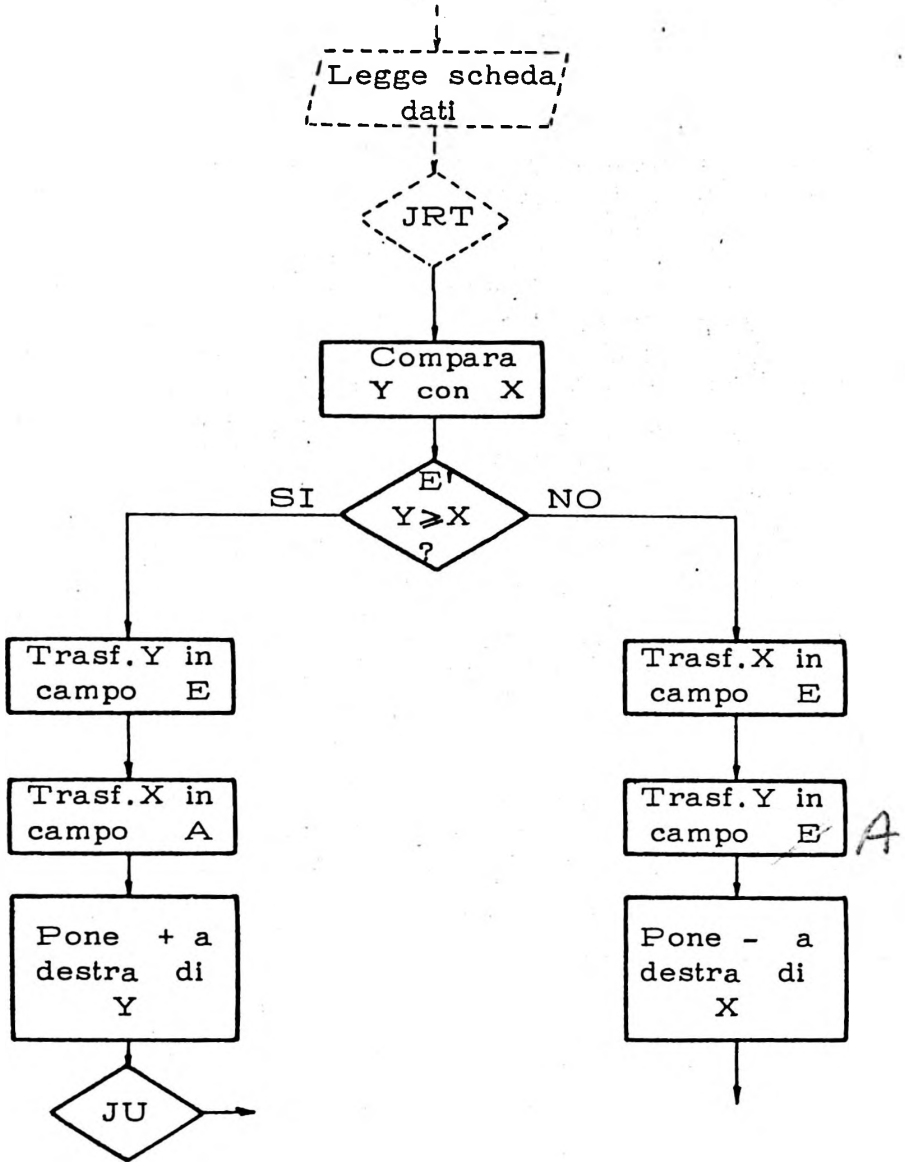
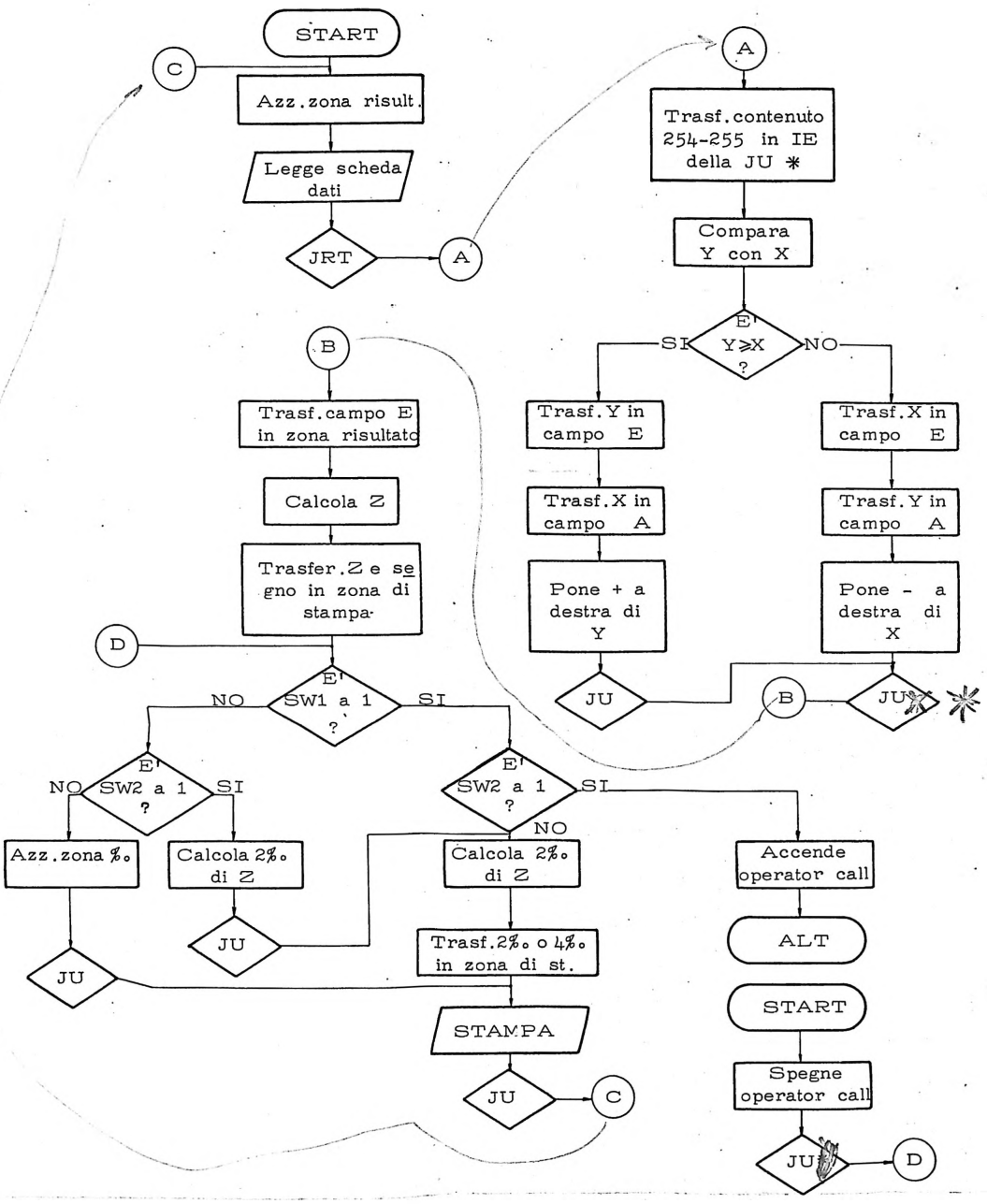

Legge scheda dati
JRT
Compara Y con X
E'
Y≥X
?
SI
NO
Trasf.Y in campo E
Trasf.X in campo A
Pone + a destra di Y
JU
Trasf.X in campo E
Trasf.Y in campo E
A
Pone - a destra di X

fig. 21.3.1



Programma : eseguire Y-X=Z

Modalità operative : SW1 a 1 e SW2 a 0 calcolare il 2% di Z
SW1 a 0 e SW2 a 1 " " 4% di Z
SW1 a 0 e SW2 a 0 non calcolare alcun %.
SW1 a 1 e SW2 a 1 arrestare l'elaborazione con la indicazione "operator call"accesa.

fig. 21.3.2

Formando un unico flow-chart con i due riportati in Fig. 21.2.2. e in Fig. 21.3.1., si ottiene il flow-chart completo del programma in esame (Fig. 21.3.2)

DOMANDE

Dopo avere attentamente esaminato il flow-chart di Fig. 21.3.2. e la codificazione delle relative istruzioni riportate sul foglio programma di Fig. 21.3.3. si risponda alle seguenti domande :

a) - Perchè la JU che ha il codice di funzione all'indirizzo 0/0 8/C non ha indicato l'indirizzo al quale deve saltare il programma ?

perché ? Viene completato dal M.U.C. ob 6/A

b) - Dopo l'istruzione " LEGGI " è stata inserita nel programma una JRT che permette di saltare sulla sub-routine segno. Si poteva usare una JU anzichè la JRT ?

Sì ... ma.... se si vuole utilizzare più volte nel proprio programma la sottrazione ... occorre memorizzare il punto di JRT...

c) - Quale zona di memoria è occupata dal programma ?

dallo 0/0 allo 0/0 A/3

d) - Si indichi, con i rispettivi indirizzi di fine ed inizio, le zone di memoria occupate dai seguenti dati:

- Z in zona risultato
- Z in zona di stampa
- segno di Z in zona di stampa
- parte millesimale di Z in zona risultato
- parte millesimale di Z in zona di stampa.

FOGLIO PROGRAMMA GE-115

F	L/F	IE+	IE-	IA+	IA-	Rif.	Codice simb.	Indir. esadec	Ind. dec.	Note
7,0	0,0						NOP2	0,0 0,0	00	
7,0	0,0						NOP2	0,0 0,2	02	
7,0	0,0						NOP2	0,0 0,4	04	
							Leggi	0,0 0,6	06	Legge scheda dati
4,5	F,0	0,0	6,8				JRT	0,0 0,A	10	Salta ad Ⓐ
D,2	0,5	0,2	A,9	0,1	9,0		MVC	0,0 0,E	14	Trasf. c.po Ein zona ris.
0,7	0,0						NOP2	0,0 1,4	20	
0,7	0,0						NOP2	0,0 1,6	22	
0,7	0,0						NOP2	0,0 1,8	24	
F,B	5,4	0,2	A,D	0,1	9,C		SD	0,0 1,A	26	Calcola Z
D,2	0,6	0,3	0,2	0,2	A,8		MVC	0,0 2,0	32	Trasf. Z in zona stampa
5,3	8,0	0,0	4,2				JS1	0,0 2,6	38	E'switch 1 inserito? Ⓓ
5,3	4,0	0,0	3,8				JS2	0,0 2,A	42	E'switch 2 inserito ?
D,7	0,4	0,2	B,0	0,2	B,0		XC	0,0 2,E	46	Azzer. zona %.
4,7	F,0	0,0	5,2				JU	0,0 3,4	52	
F,A	6,6	0,2	A,D	0,2	A,D		AD	0,0 3,8	56	Calcola il 2%.
4,7	F,0	0,0	4,6				JU	0,0 3,E	62	
5,3	4,0	0,0	5,A				JS2	0,0 4,2	66	E' switch 2 inserito?
F,A	6,6	0,2	A,D	0,2	A,D		AD	0,0 4,6	70	Calcola il 2%.
D,2	0,3	0,3	1,2	0,2	A,7		MVC	0,0 4,C	76	Trasf. %. in zona stampa
							Stampa	0,0 5,2	82	Stampa Z e %.
4,7	F,0	0,0	0,0				JU	0,0 5,6	86	Salta a Ⓒ
0,2	8,0						LON	0,0 5,A	90	Accende operator call
0,A	0,0						HLT	0,0 5,C	92	Alt err. posiz. switches
0,2	4,0						LOFF	0,0 5,E	94	Spegne operator call
4,7	F,0	0,0	2,6				JU	0,0 6,0	96	Salta a Ⓓ
0,7	0,0						NOP2	0,0 6,4	100	
0,7	0,0						NOP2	0,0 6,6	102	
0,7	0,0						NOP2	0,0 6,8	104	Ⓐ
D,2	0,1	0,0	8,E	0,0	F,E		MVC	0,0 6,A	106	Compila IE di JU*
F,9	4,4	0,2	0,4	0,2	0,9		CMQ	0,0 7,0	112	Compara Y con X
4,7	3,0	0,0	9,0			≥	JGE	0,0 7,6	118	Salta per Y≥X
D,2	0,4	0,1	9,0	0,2	0,5		MVC	0,0 7,A	122	Trasfer. Xin campo E
D,2	0,4	0,1	9,8	0,2	0,0		MVC	0,0 8,0	128	Trasfer. Yin campo A
9,2	A,A	0,1	9,5				MVI	0,0 8,6	134	Pone - a destra di X
0,7	0,0						NOP2	0,0 8,A	138	
4,7	F,0	x,x	x,x				JU*	0,0 8,C	140	Salta a Ⓑ'
D,2	0,4	0,1	9,0	0,2	0,0		MVC	0,0 9,0	144	Trasfer. Y in campo E
D,2	0,4	1,1	9,8	0,2	0,5		MVC	0,0 9,6	150	Trasfer. X in campo A
9,2	B,0	0,1	9,5				MVI	0,0 9,C	156	Pone + a destra di Y
4,7	F,0	0,0	8,A				JU	0,0 A,0	160	

fig. 21.3.3

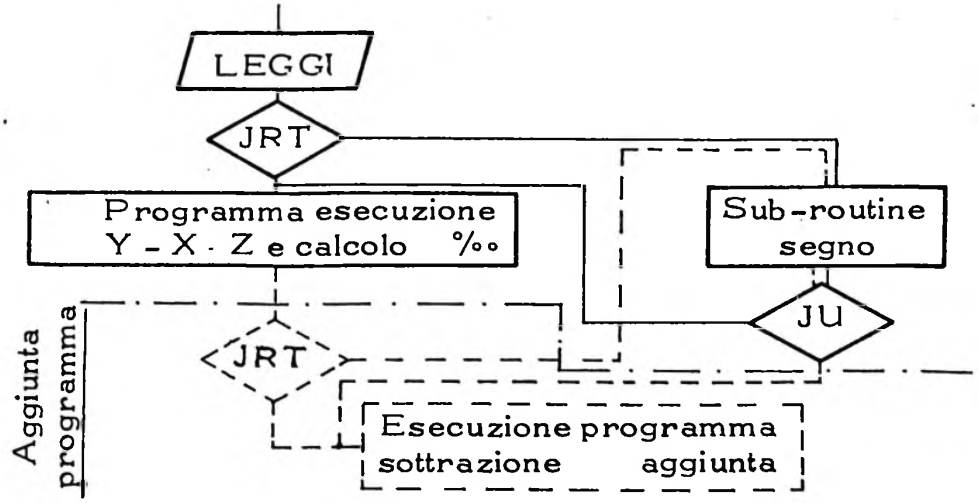
Pos. 254/255 dove si è memorizzato l'indirizzo seguente all'IRT cioè nel caso sul posto: 0,0,E

RISPOSTE

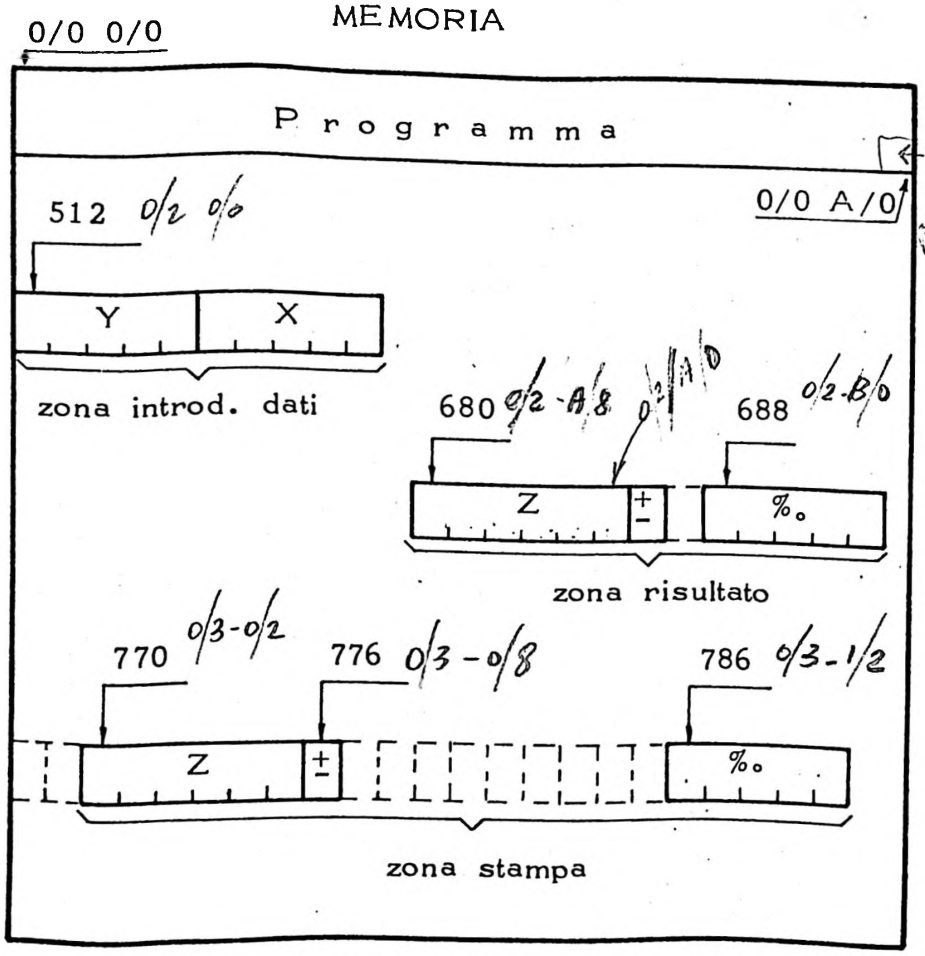
a) - Perchè viene compilato dall'istruzione MVC, con codice di funzione all'indirizzo 0/0 6/A, che trasferisce l'indirizzo memorizzato con lo scarica - mento VE della JRT, nelle posizioni di memoria 0/0 FE, 0/0 F/F (254-255). Questo permette di riprendere il programma dall'istruzione seguente la JRT, e cioè dalla MVC che ha il codice di funzione all'indirizzo 0/0, 0/E.

b) - Si poteva programmare una JU con lo stesso indirizzo di salto della JRT. In questo caso però, la MVC posta all'indirizzo 0/0 6/A, che precede la " compara Ycon X " non viene utilizzato.
In tal caso si deve codificare l'indirizzo del salto della JU che ha il codice di funzione all'indirizzo 0/0 8/C.

Riassumendo le considerazioni fatte in questa risposta e nella precedente si nota che l'uso della JRT, per il programma in esame, non è indispensabile.
Si noti in particolare molto interessante; se il programma dovesse essere ampliato per l'esecuzione di un'altra sottrazione, essenso la " sub-routine segno " organizzata nel modo visto, questa potrebbe essere utilizzata per controllare il segno di entrambe le operazioni, nel modo illustrato nel seguente schema di principio :



c, d) - Si vedano gli indirizzi indicati nella seguente figura :



	Indirizzo	
	Decimale	Esadec.
	512	0/2 0/0
1.	680	0/2 A/8
	688	0/2 B/0
2	770	0/3 0/2
3	776	0/3 0/8
5	786	0/3 1/2

LEZIONE 22

685 l.
ggco IE Rilet elev= 0/2 A/D - 6 cav. = 0/2 A/8 = 680

Tempo di studio
1 ora

elev. all'interno 0/3/02 - = 770 ---- 775
775+1 = 776 = 0/3 0/8
r/c

22. ISTRUZIONE DI TRASCODIFICA

22.1. Tale istruzione fa parte del gruppo A2 e viene qui di seguito analizzata.

22.2. TRASCODIFICA

TR (Translate Octets).

D/C L'istruzione di trascodifica permette di trasformare i caratteri configurati in un certo codice in altri corrispondenti appartenenti ad un altro codice.

Un esempio chiarirà il problema da risolvere. Si è già visto che una configurazione binaria identifica caratteri diversi in relazione al tipo di codice al quale appartiene.

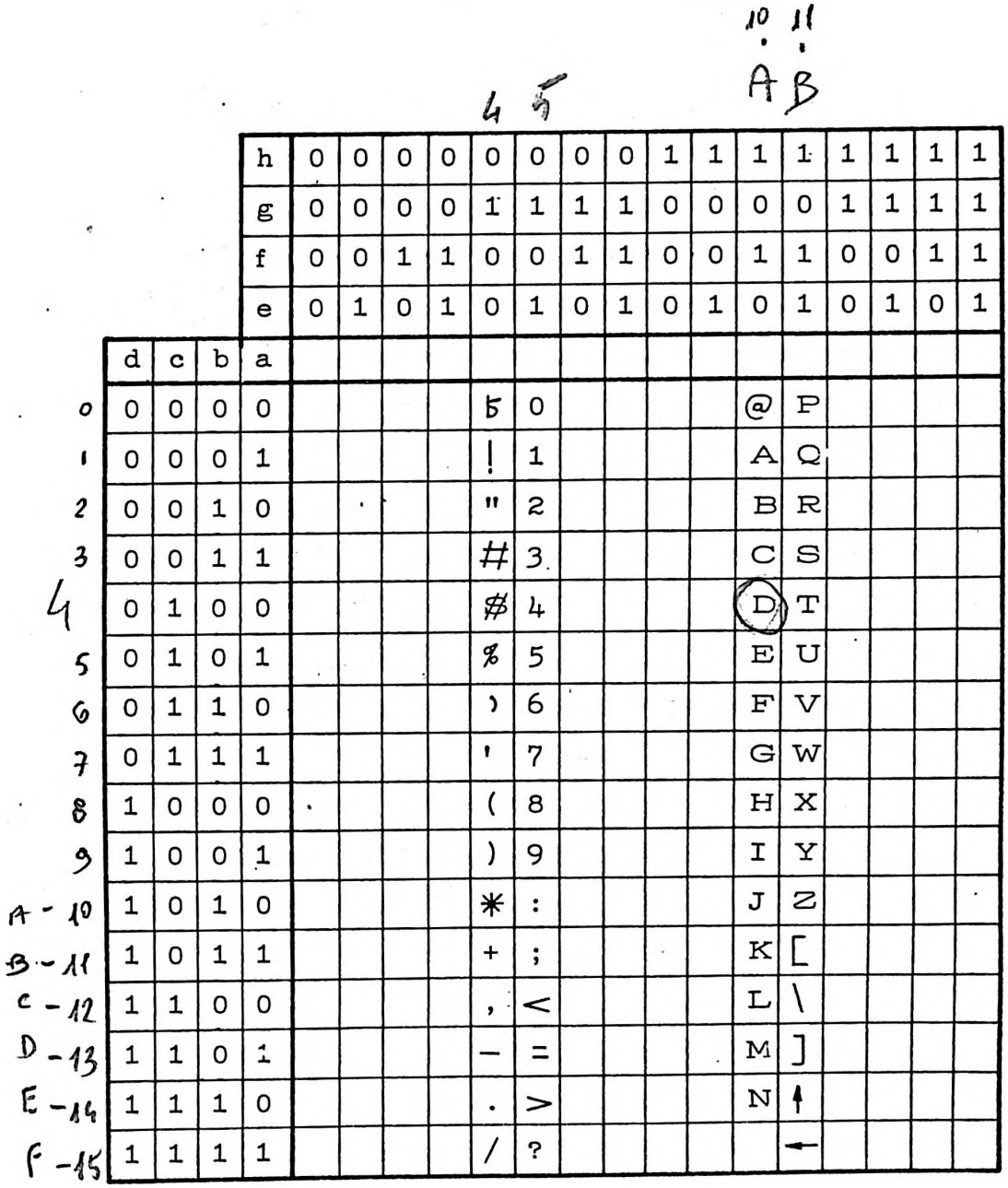
Osservando la tabella del codice GECD si può infatti notare che il carattere M è identificato dalla configurazione binaria 10100100 che espressa in valore esadecimale codificata risulta A/4. Se ora, ad esempio, si desidera stampare tale carattere con la stampante MB è necessario verificare se è possibile trasmettere al governo della stampante direttamente il carattere in codice GECD o si deve prima fare una particolare operazione (TR) per cambiare configurazione al carattere stesso.

Nelle tabelle di Figg. 22.1.1. e 22.1.2. sono riportati, rispettivamente, il codice di macchina del GE - 115 (il GECD) e quello di stampa con stampante MB.

	h	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	g	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	f	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	e	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	d c b a																
0	0 0 0 0					0	ъ					Λ	+				
1	0 0 0 1					1	A					J	/				
2	0 0 1 0					2	B					K	S				
3	0 0 1 1					3	C					L	T				
4	0 1 0 0					4	D					M	U				
5	0 1 0 1					5	E					N	V				
6	0 1 1 0					6	F					O	W				
7	0 1 1 1					7	G					P	X				
8	1 0 0 0					8	H					Q	Y				
9	1 0 0 1					9	I					R	Z				
A-10	1 0 1 0					[	&					–	\				
B-11	1 0 1 1					#	•					$	,				
C-12	1 1 0 0					_	]					*	%				
D-13	1 1 0 1					:	(					)	=				
E-14	1 1 1 0					>	<					;	"				
F-15	1 1 1 1					?	\					'	!				

fig. 22.1.1

MB (stencil)



				h	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
				g	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
				f	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
				e	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	d	c	b	a																
0	0	0	0	0					↳	0					@	P				
1	0	0	0	1					!	1					A	Q				
2	0	0	1	0					"	2					B	R				
3	0	0	1	1					#	3					C	S				
4	0	1	0	0					$	4					D	T				
5	0	1	0	1					%	5					E	U				
6	0	1	1	0					&	6					F	V				
7	0	1	1	1					'	7					G	W				
8	1	0	0	0					(	8					H	X				
9	1	0	0	1					)	9					I	Y				
A - 10	1	0	1	0					*	:					J	Z				
B - 11	1	0	1	1					+	;					K	[				
C - 12	1	1	0	0					,	<					L	\				
D - 13	1	1	0	1					-	=					M	]				
E - 14	1	1	1	0					.	>					N	↑				
F - 15	1	1	1	1					/	?						←				

fig. 22.1.2

Se si osserva il codice STAMPA MB si nota che la configurazione A/4 non corrisponde al carattere M bensì al carattere D.

Il carattere M in codice stampa MB presenta invece la configurazione binaria 11011010 A/D; ciò porta alla seguente conclusione :

- qualora si elaborino dei dati i cui risultati devono essere stampati mediante la stampante MB é necessario fare una conversione, o trascodifica, dei caratteri elaborati. Cioé é necessario far corrispondere i dati elaborati in codice GECD in configurazioni binarie che esprimono gli stessi caratteri ma in codice stampa MB. Si tratta quindi di far in modo che, come dall'esempio precedente

M (GECD)	M (St.MB)
A/4	A/D

ciò per consentire all'elaboratore d'inviare all'esterno un carattere che stampato, corrisponda ad una M (cioé il carattere elaborato) e non ad un carattere diverso (D).

DOMANDE

a) - E' possibile inviare direttamente ad una unità periferica avente codice diverso, i dati elaborati in codice macchina ? No -- bisogna transcodificarli ---

b) - A quali configurazioni binarie in codice GECD corrispondono i caratteri A, 9, Z, D, 4, 7 ?

A = 5/1 = 1.1.1 / 9 = 4/9 = 1.1.1 / Z = 11/9 = 1.1.1.1

D = 5/4 = 1.1.1 / 4 = 4/4 = 1.1.1 / 7 = 4/7 = 1.1.1

c) - A quali configurazioni binarie dal codice stampante MB corrispondono i valori A, 9, Z, D, 4, 7 ?

A = 10/1 = 1.1. / 9 = 5/9 = 1.1.1 / Z = 01/10 = 1.1.1.

D = 10/4 = 1.1. / 4 = 5/4 = 1.1.1 / 7 = 5/7 = 1.1.1

d) - Esprimere in esadecimale codificato la configurazione dei seguenti caratteri sia nel codice GECD che nel codice di stampa della MB

CARATTERE	CODICE	
	GECD	Stampa MB
A	5/1	A/1
9	4/9	5/9
Z	B/9	B/A
D	5/4	A/4
4	4/4	5/4
7	4/7	5/7

RISPOSTE

a) - No, è necessario prima "trascodificare" i singoli caratteri dal codice GECD nel codice usato dalla macchina periferica.

b) -

CODICE GECD	
A	01010001
9	01001001
Z	10111001
D	01010100
4	01000100
7	01000111

c) -

CODICE Stampa MB	
A	10100001
9	01011001
Z	10111010
D	10100100
4	01010100
7	01010111

d) -

CARATTERE	CODICE	
	GECD	Stampa MB
A	5/1	A/1
9	4/9	5/9
Z	B/9	B/A
D	5/4	A/4
4	4/4	5/4
7	4/7	5/7

22.2. Nel paragrafo precedente si è preso in considerazione il codice stampa MB e si è indicata la necessità, qualora si dovesse usare un tale codice, di trascodificare i caratteri in codice GECD nel codice di stampa MB.

Generalizzando si può affermare che ogni qualvolta è necessario convertire dei caratteri da un codice in un altro (da una configurazione binaria in un'altra) si deve agire in maniera opportuna. Il GE-115 ha una apposita istruzione, la TR, che provvede a tale operazione.

Perchè la TR possa conseguire l'effetto voluto è però indispensabile procedere secondo un certo modo nella disposizione dei dati in memoria.

Infatti l'istruzione di trascodifica si avvale di una particolarità relativa agli elaboratori che indirizzano le posizioni di memoria in modo binario.

Nel GE-115 si è già visto che gli indirizzi di memoria vengono appunto espressi in valore binario cioè come rappresentato in Fig. 22.2.1.

IE^+	IE^-	IA^+	IA^-
00001000 0/8	00011001 1/9	00000000 0/0	00001000 0/8

fig. 22.2.1

Ora, poichè anche i caratteri di un codice vengono espressi in binario si possono fare alcune considerazioni.

Ad esempio il carattere D in codice GECD, espresso in esadecimale codificato con 5/4, rappresenta una configurazione binaria che se espressa in valore decimale diventa :

5	4
0101	0100

Poiché é sempre possibile caricare un indirizzatore in maniera che l'indirizzo di memoria al quale si va ad estrarre un carattere sia 84, si può concludere che se si dà al campo A l'indirizzo 84 (5/4), tale indirizzo ha una configurazione binaria che in codice GECD corrisponde al carattere D. Così se al campo A si dà l'indirizzo 195 (B/9) esso ha una configurazione binaria che corrisponde in codice GECD al carattere Z.

In definitiva esistono dei valori degli indirizzi che corrispondono ai caratteri di un codice, pertanto una configurazione binaria può rappresentare sia un indirizzo, se usata a tale scopo, sia il carattere che appartiene ad un determinato codice.

DOMANDE

a) - Un carattere espresso in un certo codice può corrispondere ad un indirizzo ? Si

.....
.....

b) - Se si pone una configurazione binaria corrispondente ad un carattere in un registro indirizzatore essa come viene interpretata?

come indirizzo
.....
.....

c) - Nel codice GECD (vedi tabella di Fig. 22.1.1.) quanti caratteri stampabili corrispondono ad eventuali indirizzi ? -64-

.....
.....

RISPOSTE

a) - Si, se l'indirizzo è espresso in binario puro.

b) - Come un indirizzo.

c) - 64 indirizzi.

22.3. L'istruzione TR (trascodifica) si basa sul principio precedentemente descritto che utilizza la configurazione binaria di un carattere per determinare una posizione di memoria ben definita.

Sia l'indirizzo IA^+ IA^- inizialmente a valore 0/0 0/0, e nel campo E si abbia registrato il nome MARIA i cui caratteri espressi in codice GECD hanno il valore esadecimale codificato riportato nella tabella di fig. 22.3.1.

	M	A	R	I	A
CODICE GECD	A/4	5/1	A/9	5/9	5/1

fig. 22.3.1

Si voglia ora " trascodificare" tale nome in codice stampa MB cioé lo si voglia trasformare nella configurazione binaria sempre espressa in codice esadecimale codificata riportata nella tabella di fig. 22.3.2.

	M	A	R	I	A
CODICE Stampa MB	A/D	A/1	B/2	A/9	A/1

fig. 22.3.2

Se, come si vedrà in seguito, si estrae dalla memoria il carattere M (A/4) in codice GECD e lo si pone quale indirizzo di IA si ottiene che si estrarrà dalla posizione di memoria.

IA	
00000000	10100100
164	

fig. 22.3.3

un carattere.

Ora se si é avuta l'accortezza di mettere nella posizione 164 di memoria una configurazione binaria corrispondente al carattere M espresso in codice stampa MB (A/D) si estrarrà dalla memoria proprio tale configurazione.

Cioé mentre con il carattere del codice da trascodificare si va a creare l'indirizzo di memoria, da quest'ultima posizione preventivamente predisposta, si estrae una configurazione corrispondente al carattere "trascodificato" (nell'esempio la configurazione del carattere M in codice stampa MB).

Da quanto detto si può dedurre che la memoria deve essere preventivamente ed opportunamente predisposta.

Il modo di procedere é il seguente :

- si considera la configurazione binaria del carattere da trascodificare e gli si attribuisce il corrispondente valore decimale. Ad esempio se si vuol trascodificare il il codice GECD e si prende in considerazione il carattere Z si osserva che il suo valore decimale é 195. 185
- in tale posizione di memoria si introduce la configurazione binaria in codice stampa MB del carattere Z (B/A). Procedendo allo stesso modo per ogni carattere da trascodificare si creerà in memoria un campo denominato "tabella di trascodifica" nel quale sono predisposti tutti i caratteri aventi la configurazione dei caratteri corrispondenti al codice che si vuole trascodificare.

DOMANDE

a) - Nell'indirizzo determinato dal carattere che si vuol trascodificare , che configurazione si mette ?

la configurazione binaria del Nuovo carattere dell'altro codice che si vuole ottenere.

b) - Per agire con l'istruzione di trascodifica si deve preventivamente predisporre la memoria in ma - niera opportuna ?

Agli indirizzi (o posizioni di memoria) corrispondent. alle configurazioni binarie di un codice da trascodificare si pongono i nuovi caratteri o configurazioni di bit...

c) - Creare la " tabella di trascodifica " cosa significa ?

Predisporre la memoria con i caratteri voluti agli indirizzi corrispondenti alle configurazioni binarie dei caratteri da trascodificare -

RISPOSTE

a) - Si deve introdurre la configurazione corrispondente al medesimo carattere nel codice che si vuole ottenere.

b) - Si, una zona di memoria deve contenere i caratteri del codice che si vuol ottenere.

c) - Significa predisporre la memoria in maniera che ad ogni valore dei caratteri da trascdoficare corrisponda la configurazione binaria del medesimo carattere nel codice che si vuole ottenere.

22.4. Negli schemi seguenti si può osservare il modo di procedere dell'istruzione TR i cui valori dei caratteri che la compongono sono riportati nella tabella di figura 22.4.1

fig. 22.4.1

F	L	IE^+ IE^-	IA^+	IA^-
D/8	XX	IE	IA	

L'istruzione appartiene al gruppo A2 e si svolge, per ogni carattere da trascodificare, in tre tempi successivi.

Dopo aver eseguita la fase alfa si avrà :

1) tempo (vedi disegno di fig. 22.4.2)

- l'indirizzatore VE che contiene l'indirizzo iniziale dei dati da trascodificare estrae da memoria il carattere e non conta (mantiene lo stesso indirizzo) tale carattere lo pone in RO e da questo nell'indirizzatore VA.

2° tempo (vedi disegno di fig. 22.4.3)

- l'indirizzatore VA caricato, nella sua parte meno significativa, del carattere da trascodificare, indirizza la posizione di memoria dove preventivamente era stata posta la "tabella di trascodifica" ed estrae il carattere in configurazione binaria relativa al codice voluto. Il carattere estratto da memoria viene posto tramite RO in RA.

3° tempo (vedi disegno di fig. 22.4.4)

- nuovamente l'indirizzatore VE pilota la posizione di memoria precedente (in quanto non ha contato) ma il contenuto della memoria NON viene estratto per essere posto in RO (perché c'é il comando INIB. MEM. IN RO) bensì si travaserà il contenuto di RA in RO (carattere trascodificato) ed é quest'ultimo che verrà scritto in memoria nella medesima posizione nella quale si trovava, ma in codice diverso. Il contatore LO conta per indicare che si é elaborato un carattere.

Così si procede finchè la lunghezza (LO) non determina la fine dell'operazione.

Al termine di quest'ultima il campo E che conteneva il dato espresso in un codice GECD, conterrà ancora il medesimo dato espresso però in codice stampa MB.

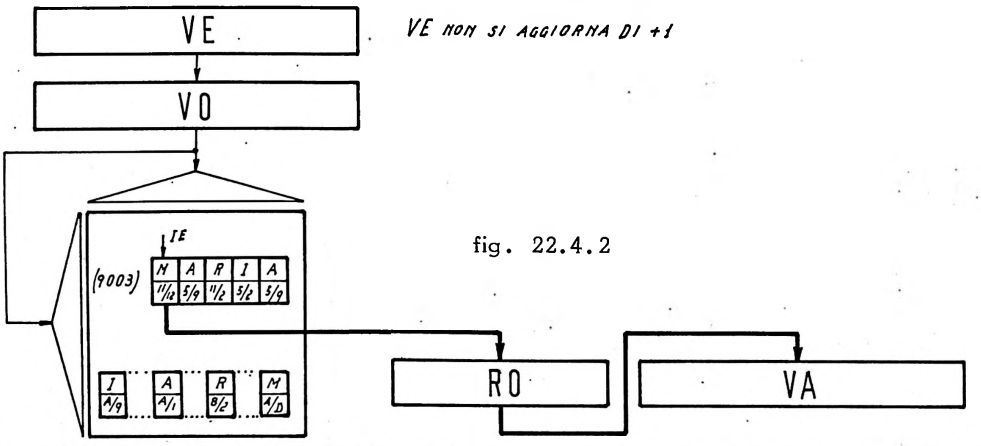


fig. 22.4.2

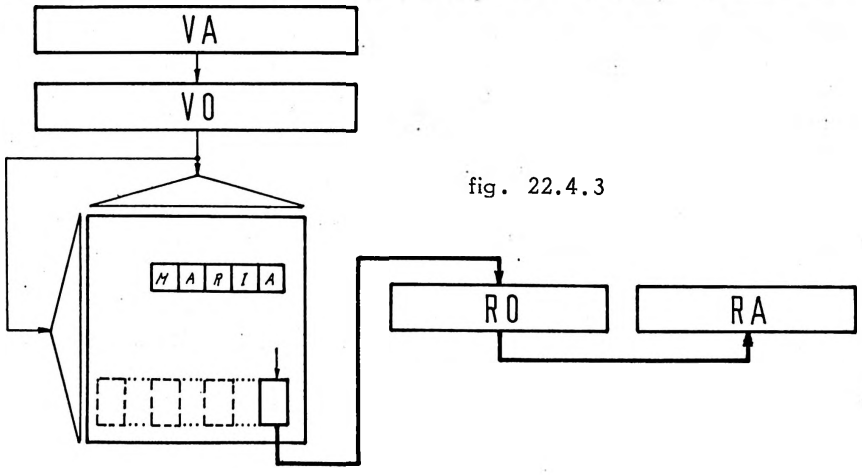


fig. 22.4.3

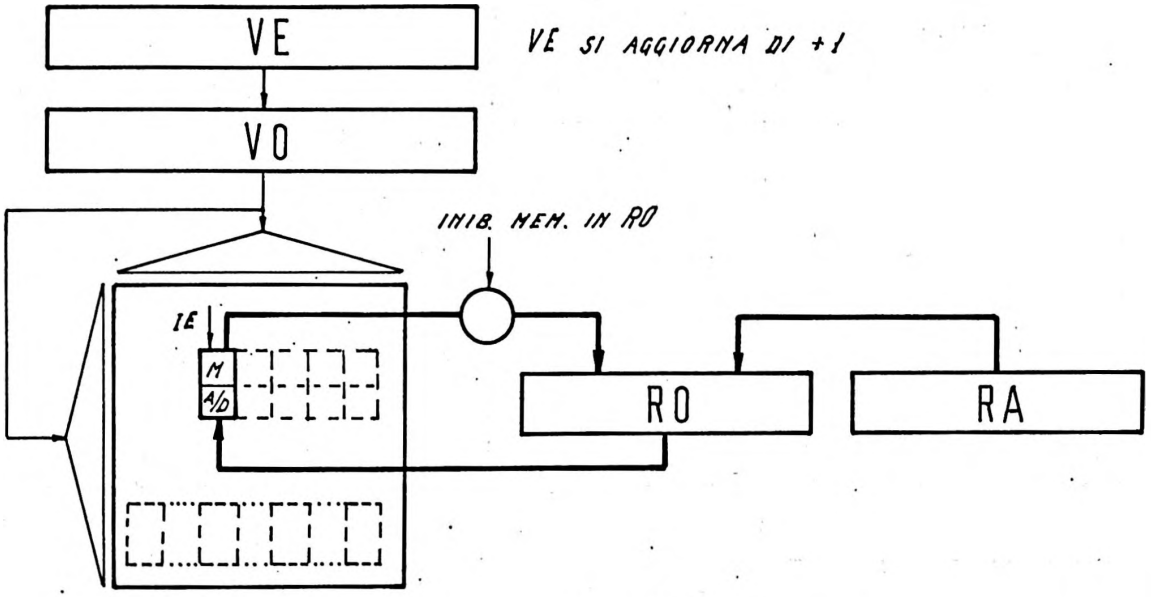


fig. 22.4.4

DOMANDE

a) - Qual'è l'indirizzatore usato per pilotare le posizioni di memoria dove è contenuto il dato da trascodificare ?

VE

b) - In quanti tempi successivi si svolge l'operazione di trascodifica ? tre tempi

c) - Nel primo tempo cosa esegue l'elaboratore ?

Estrae il carattere da trascodificare e lo pone in R.O. mette in V.A. (parte meno significativa VA)

d) - Nel secondo tempo cosa esegue l'elaboratore ?

V.A. mirata all'estrazione del carattere Nuovo (Trascodificato) e lo pone in R.O. e poi in R.A.

e) - Nel terzo tempo cosa esegue l'elaboratore ?

R.A. pone il suo contenuto in R.O. e fa in modo che venga scritta nell'indirizzo dove si era estratto il carattere da Trascodificare - IE. N. meno 2

f) - Che cosa contiene il campo A e come risulta tale campo al termine dell'operazione ?

Contiene i caratteri del Nuovo Codice (corrispondente come posizione al valore binario del vecchio codice) immutato

g) - Al termine dell'operazione cosa si trova nel campo E ?

I caratteri in lettura (ovvero trascodificati)

h) - In che parte dell'indirizzatore VA si introduce il carattere da trascodificare ?

Parte meno significativa - IA -

RISPOSTE

a) - L'indirizzatore VE.

b) - In tre tempi successivi.

c) - Estrae dalla memoria il carattere da trascodificare e lo pone nella parte meno significativa di VA.

d) - Da memoria, tramite VA, si estrae il carattere trascodificato e lo si pone in RA.

e) - Si pone il carattere trascodificato da RA in RO e da RO in memoria nella stessa posizione di memoria dalla quale si era estratto il carattere da trascodificare.

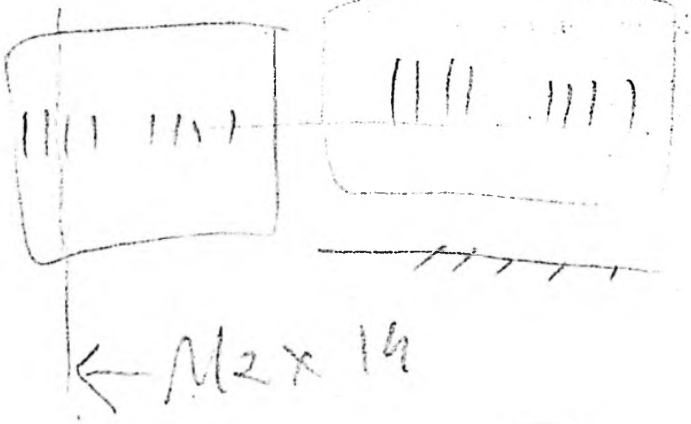
f) - La tabella di trascodifica, alla fine dell'operazione resta immutato.

g) - Il dato trascodificato.

h) - Nella parte meno significativa.

Dopo quanto osservato in precedenza si può concludere che l'indirizzo della zona nella quale é messa la tabella può variare solo di multipli di 256. Infatti durante l'operazione di trascodifica si varia solo il contenuto di VA relativo alla parte meno significativa.

La parte più significativa rimane immutata ciò significa che se tale parte ha valore 0/0 la tabella sarà posta tra gli indirizzi 0÷255 se invece la parte più significativa ha valore 0/1 la tabella sarà posta da 256 ÷ 512 e così via aumentando i valori della parte più significativa di VA di un multiplo di 256.



indirizzo Max = 256

bastano 8 bit (l'indirizzo può contenere anche 14 pos.)

F = cod. Funz.

L = Lunghezza

I.E = Indirizzo Zona Altre Aree

IA = Indirizzo (IA+) se... si vuole utilizzare zone differenti e multipli di 256 per porre la tabella nel Nuovo codice — mentre IA - è occupato dalle variazioni ...

RIEPILOGO

Tempo di studio
ore 1 e 40 minuti

1. COS'E UN PROGRAMMA ?

- un insieme di dati. (9)
- un insieme ordinato di istruzioni. (16)
- un complesso di caratteri disordinati. (7)

2. IL PROGRAMMA DEVE EFFETTUARE :

- lo svolgimento di un lavoro specifico. (4)
- la scelta tra più lavori casuali. (25)
- l'analisi casuale dei dati. (9)

3. I REGISTRI INDIRIZZATORI DELL'ELABORATORE SONO :

- VO - VI - VE - VU. (63)
- VI - VE - VA. (125)
- VO - VI - VA - VE. (82)

4. PER FAR SVOLGERE UN DETERMINATO LAVORO ALL'ELABORATORE SI DEVE INTRODURRE:

- un insieme di dati. (9)
- un programma specifico. (1)
- un programma casuale. (5)

5. PUO' UN PROGRAMMA CASUALE FAR ESEGUIRE ALL'ELABORATORE UN LAVORO DETERMINATO ?

- si. (11)
- no. (1)

6. QUANDO AGISCE L'UNITA' ARITMETICA QUALE AZIONE SI COMPIE SULLA LINEA DI RITARDO ?

- si ritarda la partenza del secondo spezzone della L.d.R. (130)
- si interrompe. (93)

7. SE L'ELABORATORE ANALIZZA COME PROGRAMMA UN COMPLESSO CASUALE DI CARATTERI PUO' SVOLGERE UN LAVORO SPECIFICO ?

- si. (2)
- no. (4)
- forse. (8)

8. CI SI PUO' AFFIDARE ALLA CASUALITA' PER SVOLGERE UN LAVORO DETERMINATO ?

- si. (25)
- no. (2)

9. IL DATO E' :

- un insieme di caratteri da elaborare. (67)
- un insieme di caratteri che non hanno nessuna importanza. (28)

10. QUAL'E' L'ORGANO CHE PERMETTE L'AGGIORNAMENTO DEGLI INDIRIZZI ?

- la rete di conta degli indirizzatori. (17)
- un organo diverso dalla rete di conta. (60)
- il registro RA. (125)

11. SI VEDANO LE LEZIONI 2 E 3.

12. DURANTE LA FASE BETA DI UN'ISTRUZIONE :

- VI non evolve. (20)
- VA e VE non possono evolvere. (44)
- nessun registro indirizzatore è interessato. (64)

13. SI VEDA LA LEZIONE 17.

14. IL REGISTRO LO DA' LA FINE ELABORAZIONE QUANDO CONTIENE TUTTI " 1 ".

- tornare alla domanda 45.

15. LE ISTRUZIONI DEL GRUPPO A1 SI DIFFERENZIANO DA QUELLE DEL GRUPPO A2:

- perchè sono costitutite da un numero diverso di caratteri. (31)
- perchè le A2 sono eseguite per indirizzi crescenti mentre le A1 sono eseguite generalmente per indirizzi decrescenti ed hanno il carattere L che ha un diverso significato. (22)
- perchè il carattere di funzione delle A1 interessa un organo diverso da quello delle A2. (100)

16. DA COSA E' FORMATA UN'ISTRUZIONE ?

- da un insieme di caratteri predisposti. (30)
- da un insieme di caratteri casuali. (8)
- da un gruppo di dati. (9)

17. QUALI OPERAZIONI PUO' SVOLGERE LA RETE DI CONTA DEGLI INDIRIZZATORI ?

- contare solo - 1. (13)
- contare $\pm$ 1. (66)

18. COME SI CHIAMANO LE FASI ?

- alfa o preparazione, gamma o di esecuzione. (124)
- beta o di preparazione, alfa o di esecuzione. (94)
- alfa o di preparazione, beta o di esecuzione. (107)

19. SI VEDANO LE LEZIONI 17,18,19,20.

20. QUALE SCOPO PUO' AVERE UN'ISTRUZIONE JU ALLA FINE DI UN PROGRAMMA ?

- di riprendere il programma dall'inizio. (48)
- di riprendere una parte del programma. (36)
- di memorizzare l'indirizzo della posizione di memoria seguente a quello che contiene l'indirizzo della stessa JU. (65)

21. SI VEDA LA LEZIONE 4.

22. IL SECONDO CARATTERE DI UN'ISTRUZIONE DEL GE-115 HA :

- sempre il significato di lunghezza (di uno o due campi). (77)
- a seconda del tipo di istruzione ha il significato di carattere lunghezza o di complemento del codice di funzione o di costante. (89)

23. E' CORRETTO CHE UN PROGRAMMA NEL SUO SVOLGIMENTO INCONTRI ED INTERPRETI DATI COME ISTRUZIONI ?

- si. (26)
- no. (34)

24. QUANDO E' INTERESSATO VI COME CONTA LA RETE DI CONTA ?

- +1. (43)
- -1. (125)

25. SI VEDA LA LEZIONE 2.

26. SI VEDA LA LEZIONE 19.

27. SI VEDANO LE LEZIONI 3 E 4.

28. POSSONO CARATTERI CHE NON HANNO NESSUNA IMPORTANZA AVERE RILEVANZA PER IL CALCOLATORE ?

- si. (90)
- no. (4)

29. COME PROCEDE L'ELABORATORE NELLA LETTURA DI UN'ISTRUZIONE ?

- per indirizzi crescenti. (113)
- per indirizzi decrescenti. (32)

30. DI QUANTI CARATTERI PUO' ESSERE COMPOSTA UN'ISTRUZIONE ?

- da un numero fisso. (47)
- da un numero qualsiasi. (70)
- da un numero variabile da 2 a 6. (51)

31. LE ISTRUZIONI DEL GRUPPO A1 HANNO LO STESSO NUMERO DI CARATTERI DI TUTTI GLI ALTRI GRUPPI ?

- si. (47)
- no. (15)

32. COME DEVE EVOLVERE L'INDIRIZZATORE VI ?

- per indirizzi crescenti. (29)
- per indirizzi decrescenti. (24)

(Mex 6.)

33. SE UN'ISTRUZIONE E' COMPOSTA DA 20 CARATTERI OCCUPA 20 POSIZIONI DI MEMORIA, E' POSSIBILE CIO' ? ma eventualmente si

- no. (27)
- si. (35)

come fare di 20 caratteri occorrono 20 posizioni di memoria !

34. COME SI POSIZIONA NORMALMENTE IL PROGRAMMA IN MEMORIA ?

- le istruzioni si pongono una adiacente all'altra. (98)
- istruzioni - dati - istruzioni. (23)

35. IL NUMERO DEI CARATTERI DI CUI E' COMPOSTA UN'ISTRUZIONE DEVE ESSERE :

- il minore possibile. (30)
- non ha importanza. (80)

Ci riferiamo alla lunghezza di istruzione nel circuit. logici del 115

36. LA RISPOSTA E' ESATTA: SI TENGA PRESENTE CHE ANCHE LA PRIMA RISPOSTA PUO' ESSERE CONSIDERATA GIUSTA. (54)

37. SI VEDANO LE LEZIONI 12 E 13.

38. QUALE LETTERA CORRISPONDE IN CODICE ESADECIMALE CODIFICATO AL VALORE 15 ?

- F. (54)
- Q. (88)

39. QUALE ORGANO CARICANO IL QUINTO ED IL SESTO CARATTERE DI UN'ISTRUZIONE ?

-VA. (3)
-VO. (87)

40. IL RIEPILOGO E' TERMINATO.

41. SI VEDA IL VOLUME DI LOGICA.

42. CI SONO ISTRUZIONI CHE OPERANO SUI DATI PER INDIRIZZI CRESCENTI ?

- si. (29)
- no. (60)

43. IL PROGRAMMA COME PROGREDISCE ?

- per indirizzi decrescenti. (125)
- per indirizzi crescenti. (42)

44. CHE COSA CONTENGONO VA E VE DURANTE UNA FASE BETA DI UN'ISTRUZIONE ?

- gli indirizzi dei caratteri che si stanno elaborando. (12)
- non é possibile determinare la loro funzione durante una fase beta. (19)

45. A COSA SERVE LA LINEA DI RITARDO ?

- a temporizzare gli eventi relativi al trattamento di un carattere. (6)
- a temporizzare ad ogni ciclata la lettura di un intero gruppo di caratteri. (119)

46. IL REGISTRO LO CONTIENE LA LUNGHEZZA ESPRESSA NELL'ISTRUZIONE E AD OGNI CARATTERE ELABORATO ESSO DEVE TENERNE CONTO. TORNARE ALLA DOMANDA 58.

47. SI VEDANO LE LEZIONI 5 ÷ 15.

48. ANCHE LA SECONDA RISPOSTA PUO' ESSERE CORRETTA. (54)

49. GLI INDIRIZZATORI DEVONO MANTENERE SEMPRE L'INDIRIZZO DEL CARATTERE CHE SI DOVRA' ELABORARE E NON I DATI. (68)

50. ANALIZZANDO I RAGGRUPPAMENTI DELLE ISTRUZIONI SI PUO' CONSIDERARE CHE TUTTE QUELLE APPARTENENTI ALLO STESSO GRUPPO ABBIANO FUNZIONI OMOGENEE ?

- si. (47)
- no. (15)

in che? ... cioè de
quali? ... siamo sempre riferite allo stesso sedo!

51. QUALI SONO I CARATTERI CHE DEFINISCONO UN'ISTRUZIONE TIPICA GE - 115 ?

- F - L - IE - IA. (52)
- F - L - IA^+ - IA^- - IE^+ - IE^-. (61)
- F - L - IE^+ - IE^- - IA^+ - IA^-. (34)

52. L'ISTRUZIONE TIPICA DEVE CONTENERE F (codice di funzione) L (codice di lunghezza) IE (indirizzo al campo E) IA (indirizzo al campo A).
COME DI ESPRIME L'INDIRIZZO DI UN CAMPO NELLE ISTRUZIONI GE - 115 ?

- con un carattere. (53)
- con due caratteri. (51)
- con più di due caratteri. (80)

53. QUANTI BIT COMPONGONO UN CARATTERE IN CODICE GE - 115 ?

- 6. (21)
- 8. (104)

54. QUALE DEI SEGUENTI CODICI ESADECIMALI E' ESATTO ?

- Q/R. (38)
- 7/F. (79)

55. 256 CONFIGURAZIONI SONO SUFFICIENTI PER SELEZIONARE TUTTE LE POSIZIONI DI MEMORIA (modulo base di 4096 posizioni) ?

- si. (78)
- no. (57)

56. A CHE COSA PUO' SERVIRE LA CONSOLE OPERATIVA ?

- ad aumentare la capacità di memoria. (92)
- a fornire delle " indicazioni " all'elaboratore. (71)

57. SI RIESCE AD INDIRIZZARE CON UN SOLO CARATTERE DEL CODICE GECD 4096 POSIZIONI DI MEMORIA ?

- si. (80)
- no. (51)

58. LA RETE DI LO PUO' CONTARE :

- -1. (50)
- ±1. (85)
- +1. (46)

59. SI VEDA LA LEZIONE 14.

60. SI VEDA LA LEZIONE 17.

61. SI VEDA LA LEZIONE 4.

62. IN QUALE REGISTRO VIENE POSTO IL QUARTO CARATTERE DI UN'ISTRUZIONE ?

- FO. (13)
- VE. (75)
- LO. (85)

63. I DUE CAMPI NECESSARI ALLA COMPILAZIONE DELL'ISTRUZIONE TIPICA COME SI CHIAMANO ?

- campo E - A. (83)
- campo A - campo programma. (27)

64. SI VEDANO LE LEZIONI 17, 18, 19, 20.

65. QUALE ISTRUZIONE DI SALTO PERMETTE DI MEMORIZZARE L'INDIRIZZO DEL CARATTERE IMMEDIATAMENTE SEGUENTE ?

- JRT. (20)
- JU. (37)

66. QUAL'E' LO SCOPO DI VO :

- ricevere gli indirizzi dagli altri indirizzatori, restituirli aggiornati di +o-1, e selezionare la memoria. (58)
- scambiare gli indirizzi, così come li riceve, con agli altri indirizzatori. (10)
- scambiare i dati con gli altri indirizzatori, ridarli aggiornati e selezionare la memoria. (49)

67. I CARATTERI DA ELABORARE SI IDENTIFICANO CON LE ISTRUZIONI ?

- si. (90)
- no. (4)

68. GLI INDIRIZZATORI QUANDO HANNO TRASFERITO IL LORO CONTENUTO IN VO SI AZZERANO. TORNARE ALLA DOMANDA 66 E RILEGGERLA ATTENTAMENTE.

69. SI VEDA IL VOLUME "MEMORIA A NUCLEI".

70. PUO' UN'ISTRUZIONE GE - 115 ESSERE COMPOSTA DA 20 CARATTERI ?

- si. (33)
- no. (30) —

71. COME SONO UTILIZZATE DALL'ELABORATORE LE INDICAZIONI FORNITEGLI DA CONSOLE ?

- con istruzioni. (79)
- con l'accensione di opportune segnalazioni luminose. (59)

72. COME SI CHIAMA L'ORGANO CHE MEMORIZZA IL PRIMO CARATTERE DI UN'ISTRUZIONE ?

- registro FO. (123)
- registro RA. (100)

73. LA REGISTRAZIONE DI DATI IN MEMORIA, SENZA CHE QUESTI SIANO INTRODOTTI DA SUPPORTI ESTERNI :

- é possibile con opportune istruzioni interne. (116)
- non é possibile. (127)

74. SI VEDA LA LEZIONE 10.

75. COSA FORMANO IL QUINTO E SESTO CARATTERE DI UN'ISTRUZIONE ?

- l'indirizzo del campo A. (39)
- l'indirizzo del programma. (27)

76. IL REGISTRO RO QUALI CARATTERI DI MEMORIA PUO' RICEVERE ?

- tutti. (108)
- alcuni particolari caratteri. (110)

77. SI VEDANO LE LEZIONI 5 ÷ 15 .

78. SI VEDA LA LEZIONE 4.

79. E' POSSIBILE MODIFICARE LO SVOLGIMENTO DI UN PROGRAMMA CON " ORDINI " MANUALI QUANDO E' GIA' INTRODOTTO IN MEMORIA ?

- si. (96)
- no. (56)

80. SI VEDA LA LEZIONE 4.

81. IN FASE ALFA IL REGISTRO RA ASSUME LA FUNZIONE DI :

- registro ausiliario. (10)
- registro di funzione. (100)

82. QUALE ORGANO CARICANO IL TERZO E QUARTO CARATTERE ?

- VA. (75)
- VO. (83)
- VE. (66)

83. VO E' L'INDIRIZZATORE A DIRETTO CONTATTO CON I CIRCUITI DI SELEZIONE DI MEMORIA, PUO' RICEVERE UN INDIRIZZO DI PROGRAMMA ?

- si. (84)
- no. (126)

84. QUANTI CAMPI BISOGNA STABILIRE IN UN'ISTRUZIONE TIPICA ?

- 2. (3)
- più di due. (80)

85. LO E' UN REGISTRO PARTICOLARE CHE PUO' CONTENERE ?

- la lunghezza dei dati. (101)
- l'indirizzo di programma. (125)

86. CHE FUNZIONE HA IL REGISTRO RO ?

- è il registro di memoria. (76)
- è il registro indirizzatore. (125)

87. SI VEDANO LE LEZIONI 3 E 4.

88. IL CODICE ESADECIMALE PUO' ESSERE UTILIZZATO CON CODICI A :

- 3 bit. (80)
- 8 bit. (103)

89. NEL GE-115 ESISTONO ISTRUZIONI CHE PERMETTONO :

- di eseguire esclusivamente elaborazione di dati. (77)
- di eseguire una elaborazione dei dati, mentre altre istruzioni sono utilizzate per ottenere soltanto delle indicazioni senza che i dati siano interessati. (105)
- di svolgere qualsiasi tipo di elaborazione perchè ogni istruzione non ha uno scopo prefissato. (91)

90. SI VEDA LA LEZIONE 2.

91. I GRUPPI DI ISTRUZIONI SONO STATI CREATI:

- casualmente. (27)
- in funzione del numero di caratteri che le definiscono (89).

92. SI VEDA LA LEZIONE 14.

93. SE SI INTERROMPE LA LINEA DI RITARDO LO ELABORATORE SI FERMA ?

- si. (6)
- no. (125)

94. L'ISTRUZIONE VIENE SVOLTA DALL'ELABORATORE IN DUE TEMPI SUCCESSIVI CHIAMATI FASI. LA PRIMA FASE CHE PREDISPONE GLI ORGANI INTERNI COME SI CHIAMA ?

- alfa. (95)
- beta. (125)
- gamma (124)

95. LA SECONDA FASE E' QUELLA CHE ESEGUE L'OPERAZIONE PER LA QUALE L'ISTRUZIONE E' STATA CREATA : COME SI CHIAMA ?

- beta. (18)
- alfa. (60)

96. NELLE ISTRUZIONI DI CONFRONTO, IL RISULTATO E' DISPONIBILE :

- in memoria nella zona del campo E. (115)
- in opportuni organi che devono essere interrogati con un'istruzione di salto. (73)

97. SI VEDA LA LEZIONE 10.

98. E' NECESSARIO PORRE DUE CARATTERI PER STABILIRE UN INDIRIZZO DI UN CAMPO ?

- si. (29)
- no. (53)

99. IL REGISTRO INDIRIZZATORE QUANTE POSIZIONI DI MEMORIA PUO' SELEZIONARE IN UNA VOLTA ?

- tre. (102)
- tutte. (69)
- una. (45)

100. CHE SCOPO HA IL REGISTRO DI FUNZIONE ?

- memorizza il primo carattere di una istruzione. (72)
- memorizza il quinto carattere di una istruzione. (125)

101. CHE SCOPO HA IL REGISTRO DI LUNGHEZZA ?

- memorizza il settimo carattere dell'istruzione. (30)
- memorizza il quarto carattere dell'istruzione. (60)
- memorizza il secondo carattere dell'istruzione. (123)

102. CON IL CONTENUTO DEL REGISTRO INDIRIZZATORE QUALI ORGANI SI VANNO AD INTERESSARE ?

- i circuiti di selezione delle posizioni di memoria. (69)
- i circuiti dell'U.A. (125)

103. QUANTI BIT SONO UTILIZZATI PER ESPRIMERE UN CARATTERE IN ESADECIMALE ?

- 4. (109)
- 8. (61)

104. CON OTTO BIT QUANTE CONFIGURAZIONI SI POSSONO OTTENERE ?

- 256. (55)
- 32. (41)

105. IN QUANTE FASI SI SVILUPPA GENERALMENTE L'OPERAZIONE CHE UNA DATA ISTRUZIONE DEVE ESEGUIRE ?

- 2. (18)
- 4. (110)
- 6. (124)

106. SI VEDANO LE LEZIONI 6 E 10.

107. CHE COSA COMPIE L'ELABORATORE DURANTE LA FASE ALFA ?

- legge carattere per carattere l'istruzione. (45)
- legge un carattere si ed uno no. (118)
- legge casualmente i caratteri dell'istruzione. (5)

108. PUO' IL REGISTRO RO CONTENERE CONTEMPORANEAMENTE DUE CARATTERI ?

- si. (124)
- no. (114)

109. CON QUATTRO BIT CHE VALORE DECIMALE MASSIMO SI PUO' ESPRIMERE ?

- 15. (38)
- 8. (41)

110. SI VEDA LA LEZIONE 16.

111. LA RISPOSTA E' ESATTA. L'ISTRUZIONE PUO' ESSERE LA SL OPPURE LA SR. (40).

112. QUALE FUNZIONE HA L'ISTRUZIONE SL ?

- di cercare nel campo indicato dall'indirizzo IE il carattere uguale a quello posto all'indirizzo IA. (117)
- di cercare, procedendo verso sinistra, l'inizio del programma. (122)

113. OGNI CARATTERE DI UN'ISTRUZIONE VIENE POSTO IN UN ORGANO PARTICOLARE ?

- si. (72)
- no. (60)

114. DOVE VIENE MANTENUTO IL TERZO CARATTERE ESTRATTO NELLA FASE ALFA ?

- in RO. (86)
- in VE. (62)
- in LO. (85)

115. SI VEDANO LE LEZIONI 6, 8 e 10.

116. QUAL'E' L'ISTRUZIONE CHE CONSENTE DI SCRIVERE UN CARATTERE, IN UNA ZONA PRESTABILITA DI MEMORIA ?

- MVI. (129)
- CMI. (97)

117. COME E' POSSIBILE SAPERE SE DURANTE UNA ISTRUZIONE DI RICERCA (SL o SR) E' STATO TROVATO IL CARATTERE CERCATO ?

- con una successiva istruzione di salto. (128)
- con l'accensione della lampadina SL (o SR) sulla console operativa. (106)

118 NELLA FASE ALFA DI UN'ISTRUZIONE L'ELABORATORE DEVE PREDISPORSI PER LA SUCCESSIVA ESECUZIONE. QUALI SONO I CARATTERI NECESSARI A TALE PREDISPOSIZIONE ?

- i caratteri dell'istruzione letti ordinatamente. (107)
- i caratteri qualsiasi letti ordinatamente. (7)
- i dati opportunamente predisposti. (9)

119. COME VENGONO LETTI I CARATTERI ?

- tutti assieme. (120)
- uno per volta. (45)

120. QUAL'E' L'ORGANO CHE STABILISCE LA POSIZIONE DALLA QUALE PRELEVARE UN CARATTERE ?

- il registro di funzione. (100)
- il registro indirizzatore. (99)

121. E' POSSIBILE PROGRAMMARE UN CERTO NUMERO DI MVI CONSECUTIVE PER FORMARE UN DATO IN MEMORIA ?

- si. (73)
- no. (74)

122. SI VEDA LA LEZIONE 6.

123. COME SI CHIAMA L'ORGANO CHE MEMORIZZA IL SECONDO CARATTERE DELL'ISTRUZIONE ?

- registro RO. (101)
- registro LO. (114)

124. SI VEDA LA LEZIONE 16.

125. SI VEDA LA LEZIONE 17.

126. SI VEDA LA LEZIONE 4.

127. CON UN'ISTRUZIONE MVI (trasferisci l'immediato) SI PUO' :

- scrivere nella posizione di memoria indicata da IE^+ e IE^- il secondo carattere dell'istruzione stessa. (121)
- trasferire un solo carattere, dal campo A al campo E, immediatamente. (74)

128. LA RISPOSTA E' ESATTA, SI RITORNI ALLA DOMANDA 129.

129. E' POSSIBILE SAPERE, CON UN'ISTRUZIONE, SEGUITA DA UN'ISTRUZIONE DI SALTO DA UN ALT CARATTERISTICO, SE IN UN CAMPO DI 100 CARATTERI ESISTE IL CARATTERE X ?

- si. (111)
- no. (112)

130. QUALE SCOPO HA IL RITARDO DELL'AVVIO DEL SECONDO SPEZZONE DELLA LINEA DI RITARDO ?

- consentire il caricamento di RO con il carattere che provenie da memoria. (60)
- consentire a RO di ricevere un nuovo carattere da scrivere in memoria. (12)

CORSI DI ISTRUZIONE PROGRAMMATA

INTRODUZIONE AGLI ELABORATORI ELETTRONICI

1. Transistori
2. Logica
3. Memoria a nuclei
4. Oscilloscopio

SISTEMA GE - 115

1. Introduzione
2. Unità Centrale vol. 1°
3. Memoria 16 K
4. Unità Centrale vol. 2°
5. Unità Centrale vol. 3°
6. G.U.N. 7
7. Governo MZ4 - 300

M.T.E.

1. Il calcolatore M.T.E. - 2

GENERAL ELECTRIC *è marchio registrato della GENERAL ELECTRIC COMPANY U.S.A.*